



65A/1168/CDV

COMMITTEE DRAFT FOR VOTE (CDV)

PROJECT NUMBER: IEC 61508-7 ED3	
DATE OF CIRCULATION: 2025-02-14	CLOSING DATE FOR VOTING: 2025-05-09
SUPERSEDES DOCUMENTS: 65A/1062A/CD, 65A/1081A/CC	

IEC SC 65A : SYSTEM ASPECTS	
SECRETARIAT: United Kingdom	SECRETARY: Ms Stephanie Lavy
OF INTEREST TO THE FOLLOWING COMMITTEES: TC 8,TC 9,TC 22,TC 31,TC 44,TC 45,TC 56,TC 61,TC 62,TC 65,SC 65B,SC 65C,SC 65E,TC 66,TC 77,TC 80,TC 108,SyC AAL,SyC SM,SC 41	HORIZONTAL FUNCTION(S):
ASPECTS CONCERNED: Safety	
☒ SUBMITTED FOR CENELEC PARALLEL VOTING Attention IEC-CENELEC parallel voting The attention of IEC National Committees, members of CENELEC, is drawn to the fact that this Committee Draft for Vote (CDV) is submitted for parallel voting. The CENELEC members are invited to vote through the CENELEC online voting system.	☐ NOT SUBMITTED FOR CENELEC PARALLEL VOTING

This document is still under study and subject to change. It should not be used for reference purposes.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Recipients of this document are invited to submit, with their comments, notification of any relevant "In Some Countries" clauses to be included should this proposal proceed. Recipients are reminded that the CDV stage is the final stage for submitting ISC clauses. (SEE [AC/22/2007](#) OR [NEW GUIDANCE DOC](#)).

TITLE: Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 7: Overview of techniques and measures
--

PROPOSED STABILITY DATE: 2028

NOTE FROM TC/SC OFFICERS:

Copyright © 2025 International Electrotechnical Commission, IEC. All rights reserved. It is permitted to download this electronic file, to make a copy and to print out the content for the sole purpose of preparing National Committee positions. You may not copy or "mirror" the file or printed version of the document, or any part of it, for any other purpose without permission in writing from IEC.

CONTENTS

1		
2		
3	FOREWORD.....	8
4	INTRODUCTION.....	10
5	1 Scope.....	12
6	2 Normative references	14
7	3 Definitions and abbreviations.....	14
8	Annex A (informative) Overview of techniques and measures for E/E/PE safety-related	
9	systems: control of random hardware failures (see IEC 61508-2).....	15
10	A.1 Electric	15
11	A.1.1 Failure detection by on-line monitoring	15
12	A.1.2 Monitoring of relay contacts	15
13	A.1.3 Comparator	15
14	A.1.4 Majority voter.....	16
15	A.1.5 Idle current principle (de-energised to trip)	16
16	A.2 Electronic.....	16
17	A.2.1 Tests by redundant hardware	16
18	A.2.2 Dynamic principles	16
19	A.2.3 Standard test access port and boundary-scan architecture	17
20	A.2.4 (Not used)	17
21	A.2.5 Monitored redundancy	17
22	A.2.6 Electrical/electronic components with automatic check	17
23	A.2.7 Analogue signal monitoring.....	18
24	A.2.8 De-rating	18
25	A.3 Processing units	18
26	A.3.1 Self-test by software: limited number of patterns (one-channel).....	18
27	A.3.2 Self-test by software: walking bit (one-channel)	18
28	A.3.3 Self-test supported by hardware (one-channel).....	19
29	A.3.4 Coded processing (one-channel)	19
30	A.3.5 Reciprocal comparison by software	19
31	A.4 Invariable memory ranges.....	19
32	A.4.1 Word-saving multi-bit redundancy (for example ROM monitoring with a	
33	modified Hamming code)	19
34	A.4.2 Modified checksum	20
35	A.4.3 Signature of one word (8-bit)	20
36	A.4.4 Signature of a double word (16-bit).....	20
37	A.4.5 Block replication (for example double ROM with hardware or software	
38	comparison).....	20
39	A.5 Variable memory ranges	21
40	A.5.1 RAM test "checkerboard"	21
41	A.5.2 RAM test "walkpath"	21
42	A.5.3 RAM test "galpat" or "transparent galpat".....	21
43	A.5.4 RAM test "Abraham"	22
44	A.5.5 One-bit redundancy (for example RAM monitoring with a parity bit)	22
45	A.5.6 RAM monitoring with a modified Hamming code, or detection of data	
46	failures with error-detection-correction codes (EDC)	22
47	A.5.7 Double RAM with hardware or software comparison and read/write test.....	22
48	A.5.8 RAM test "march"	23



49	A.6	I/O-units and interfaces (external communication).....	23
50	A.6.1	Test pattern	23
51	A.6.2	Code protection	23
52	A.6.3	Multi-channel parallel output.....	24
53	A.6.4	Monitored outputs.....	24
54	A.6.5	Input comparison/voting.....	24
55	A.7	Data paths (internal communication).....	24
56	A.7.1	One-bit hardware redundancy.....	24
57	A.7.2	Multi-bit hardware redundancy.....	24
58	A.7.3	Complete hardware redundancy	25
59	A.7.4	Inspection using test patterns	25
60	A.7.5	Transmission redundancy	25
61	A.7.6	Information redundancy	25
62	A.8	Power supply	25
63	A.8.1	Overvoltage protection with safety shut-off	25
64	A.8.2	Voltage control (secondary).....	26
65	A.8.3	Power-down with safety shut-off	26
66	A.9	Temporal and logical program sequence monitoring	26
67	A.9.1	Watch-dog with separate time base without time-window.....	26
68	A.9.2	Watch-dog with separate time base and time-window	26
69	A.9.3	Logical monitoring of program sequence.....	26
70	A.9.4	Combination of temporal and logical monitoring of program sequences	27
71	A.9.5	Temporal monitoring with on-line check	27
72	A.10	Ventilation and heating	27
73	A.10.1	Temperature sensor	27
74	A.10.2	Fan control	27
75	A.10.3	Actuation of the safety shut-off via thermal fuse.....	27
76	A.10.4	Staggered message from thermo-sensors and conditional alarm.....	27
77	A.10.5	Connection of forced-air cooling and status indication	28
78	A.11	Communication and mass-storage	28
79	A.11.1	Separation of electrical energy lines from information lines.....	28
80	A.11.2	Spatial separation of multiple lines	28
81	A.11.3	Design for immunity to electromagnetic interference.....	28
82	A.11.4	Antivalent signal transmission.....	29
83	A.12	Sensors	29
84	A.12.1	Reference sensor	29
85	A.12.2	Positive-activated switch	29
86	A.13	Final elements (actuators)	30
87	A.13.1	Monitoring	30
88	A.13.2	Cross-monitoring of multiple actuators.....	30
89	A.14	Measures against the physical environment.....	30
90	Annex B (informative)	Overview of techniques and measures for E/E/PE safety related	
91		systems: avoidance of systematic failures (see IEC 61508-2 and IEC 61508-3).....	31
92	B.1	General measures and techniques	31
93	B.1.1	Project management.....	31
94	B.1.2	Documentation	32
95	B.1.3	Separation of E/E/PE system safety functions from non-safety functions	33
96	B.1.4	Diverse hardware	33
97	B.1.5	Traceability.....	

98	B.1.6	Functional Safety Assurance Role Independence	34
99	B.2	E/E/PE system design requirements specification	36
100	B.2.1	Structured specification	36
101	B.2.2	Formal methods.....	37
102	B.2.3	Semi-formal methods.....	38
103	B.2.4	Computer-aided specification tools	40
104	B.2.5	Checklists.....	41
105	B.2.6	Inspection of the specification.....	42
106	B.3	E/E/PE system design and development	43
107	B.3.1	Observance of guidelines and standards	43
108	B.3.2	Structured design	43
109	B.3.3	Use of well-tried components.....	43
110	B.3.4	Modularisation	44
111	B.3.5	Computer-aided design tools	44
112	B.3.6	Simulation	45
113	B.3.7	Inspection (reviews and analysis)	45
114	B.3.8	Walk-through	45
115	B.4	E/E/PE system operation and maintenance procedures	46
116	B.4.1	Operation and maintenance instructions	46
117	B.4.2	User friendliness.....	46
118	B.4.3	Maintenance friendliness	47
119	B.4.4	Limited operation possibilities.....	47
120	B.4.5	Operation only by skilled operators.....	47
121	B.4.6	Protection against operator mistakes	48
122	B.4.7	(Not used)	48
123	B.4.8	Modification protection.....	48
124	B.4.9	Input acknowledgement	48
125	B.5	E/E/PE system integration	48
126	B.5.1	Functional testing	49
127	B.5.2	Black-box testing	49
128	B.5.3	Statistical testing	50
129	B.5.4	Field experience	50
130	B.6	E/E/PE system safety validation.....	51
131	B.6.1	Functional testing under environmental conditions.....	51
132	B.6.2	Electromagnetic interference immunity testing	52
133	B.6.3	Static analysis	52
134	B.6.4	Dynamic analysis and testing	53
135	B.6.5	Failure analysis	53
136	B.6.6	Worst-case analysis	59
137	B.6.7	Expanded functional testing.....	60
138	B.6.8	Worst-case testing.....	60
139	B.6.9	Fault insertion testing	60
140	Annex C (informative)	Overview of techniques and measures for achieving systematic	
141		capability for software (see IEC 61508-3)	62
142	C.1	General.....	62
143	C.2	Requirements and detailed design	62
144	C.2.1	Structured methods	62
145	C.2.2	Data flow diagrams	64
146	C.2.3	Structure diagrams	

147	C.2.4	Not used	66
148	C.2.5	Defensive programming	66
149	C.2.6	Design and coding standards	67
150	C.2.7	Structured programming	71
151	C.2.8	Information hiding/encapsulation	72
152	C.2.9	Modular approach	72
153	C.2.10	Use of trusted/verified software elements	73
154	C.2.11	Not Used	75
155	C.2.12	Aim: To maintain consistency between lifecycle stages.	75
156	C.2.13	Stateless software design (or limited state design)	76
157	C.2.14	Offline numerical analysis	77
158	C.2.15	Message sequence charts	77
159	C.3	Architecture design	77
160	C.3.1	Fault detection and diagnosis	77
161	C.3.2	Error detecting and correcting codes	78
162	C.3.3	Failure assertion programming	78
163	C.3.4	Diverse monitor	79
164	C.3.5	Software diversity (diverse programming)	80
165	C.3.6	Backward recovery	80
166	C.3.7	Re-try fault recovery mechanisms	81
167	C.3.8	Graceful degradation	81
168	C.3.9	Not Used	82
169	C.3.10	Dynamic reconfiguration	82
170	C.3.11	Safety and Performance in real time: Time-Triggered Architecture.....	82
171	C.3.12	UML	83
172	C.4	Development tools and programming languages	85
173	C.4.1	Strongly typed programming languages	85
174	C.4.2	Language subsets	85
175	C.4.3	Not used	85
176	C.4.4	Not used	86
177	C.4.5	Suitable programming languages	86
178	C.4.6	Automatic software generation.....	88
179	C.4.7	Test management and automation tools.....	89
180	C.5	Verification and modification	89
181	C.5.1	Probabilistic testing	89
182	C.5.2	Data recording and analysis	90
183	C.5.3	Interface testing.....	90
184	C.5.4	Boundary value analysis	91
185	C.5.5	Error guessing	91
186	C.5.6	Error seeding.....	92
187	C.5.7	Equivalence classes and input partition testing	92
188	C.5.8	Structure-based testing.....	93
189	C.5.9	Control flow analysis.....	94
190	C.5.10	Data flow analysis	94
191	C.5.11	Symbolic execution.....	95
192	C.5.12	Formal proof (verification).....	95
193	C.5.13	Complexity metrics	96
194	C.5.14	Formal inspections	97
195	C.5.15	Walk-through (software)	

196	C.5.16	Design review.....	98
197	C.5.17	Prototyping/animation.....	99
198	C.5.18	Process simulation.....	99
199	C.5.19	Performance requirements.....	100
200	C.5.20	Performance modelling.....	100
201	C.5.21	Avalanche/stress testing.....	101
202	C.5.22	Response timing and memory constraints.....	101
203	C.5.23	Impact analysis.....	102
204	C.5.24	Software configuration management.....	102
205	C.5.25	Regression validation.....	103
206	C.5.26	Animation of specification and design.....	103
207	C.5.27	Model based testing (test case generation).....	104
208	C.6	Functional safety assessment.....	106
209	C.6.1	Decision tables (truth tables).....	106
210	C.6.2	Software failure analysis.....	106
211	C.6.3	Common cause failure analysis.....	115
212	C.6.4	Reliability block diagrams.....	115
213	C.6.5	V&V Methods and Techniques Supporting IEC61508-3 CI 7.3.2.3.....	116
214	Table C.6.5 : V&V Methods and Techniques supporting Part 3 Clause 7.3.2.3.....		117
215	Annex D (informative) Statistical evaluation techniques for in service software elements.....		119
216			
217	D.1	Introduction.....	119
218	D.2	Theoretical aspects.....	119
219	D.2.1	General.....	119
220	D.2.2	References.....	119
221	D.2.3	Terms and Definitions.....	119
222	D.2.4	Estimates of Software Statistical Parameters.....	120
223	D.2.5	The Sum Total of Operational Data Required.....	122
224	D.2.6	Required Conditions for Operational History to be an Effective Guide to	
225		Future Use.....	123
226	D.3	Deployment aspects.....	123
227	D.3.1	Statistical evaluation deployment guidance.....	123
228	D.3.2	Some Examples of Difficulties with Software Statistical Modelling.....	125
229	D.4	Route 2s proven in use claim.....	127
230	D.5	Bibliography.....	127
231	Annex E (informative) Overview of techniques and measures for design of complex integrated circuits.....		129
232			
233	Annex F (informative) Definitions of properties of software lifecycle phases.....		130
234	Annex G (informative) Guidance for the development of safety-related object oriented software.....		136
235			
236	Annex H (informative) NOTE The overview of techniques and measures contained in this annex has been moved ot IEC 61508-2-1.....		138
237			
238	Bibliography.....		139
239	Index.....		142
240			
241	Figure 1 – Overall framework of IEC 61508.....		13
242	Figure C.6.2.1 – Typical (qualitative) software functional fault tree.		110
243			

244	Table B.1 – Significance of Independence Concern vs. Functional Safety Assurance	
245	Activity.....	35
246	Table B.2 – Effectiveness of Independence Measure vs. Independence Concern.....	35
247	Table B.3 – Effectiveness of Independence Measure vs. Functional Safety Assurance	
248	Activity.....	36
249	Table C.1 – Recommendations for specific programming languages	87
250	Table C.6.2.1 - Example of the Guidewords as used in SHARD	110
251	Table C.6.2.2 - Example of the Guidewords as used in SHARD	113
252	Table D.1 – Failure-free observations required to reach confidence in a failure	
253	probability per demand	121
254	Table D.2 – Failure-free hours of observation required to reach confidence in a failure	
255	probability per hour.....	122
256	Table F.1 – Software Safety Requirements Specification	130
257	Table F.2 – Software design and development: software architecture design	131
258	Table F.2 – Software design and development: support tools and programming	
259	language.....	132
260	Table F.2 – Software design and development: detailed design	132
261	Table F.2 – Software design and development: software module testing and integration.....	133
262	Table F.2 – Programmable electronics integration (hardware and software).....	133
263	Table F.2 – Software aspects of system safety validation	134
264	Table F.2 – Software modification	134
265	Table F.2 – Software verification.....	135
266	Table F.2 – Functional safety assessment	135
267	Table G.1 – Object Oriented Software Architecture	136
268	Table G.2 – Object Oriented Detailed Design.....	137
269	Table G.2 – Some Oriented Detailed terms	137
270		
271		

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**FUNCTIONAL SAFETY OF ELECTRICAL/ELECTRONIC/
PROGRAMMABLE ELECTRONIC SAFETY-RELATED SYSTEMS –****Part 7: Overview of techniques and measures****FOREWORD**

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as “IEC Publication(s)”). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) IEC draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). IEC takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, IEC had not received notice of (a) patent(s), which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at <https://patents.iec.ch>. IEC shall not be held responsible for identifying any or all such patent rights.

IEC 61508-7 has been prepared by subcommittee 65A: System aspects, of IEC technical committee 65: Industrial-process measurement, control and automation.

This third edition cancels and replaces the second edition published in 2010. This edition constitutes a technical revision.

This edition has been subject to a thorough review and incorporates many comments received at the various revision stages and:

- the contents of annex E have been moved to IEC 61508-2-1;
- A revision of Annex D covering proven in use to include new wording, explanations and examples.

326 The text of this International Standard is based on the following documents:

Draft	Report on voting
XX/XX/FDIS	XX/XX/RVD

327
328 Full information on the voting for its approval can be found in the report on voting indicated in
329 the above table.

330 The language used for the development of this International Standard is English.

331 This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in
332 accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available
333 at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are
334 described in greater detail at www.iec.ch/publications.

335 A list of all parts of the IEC 61508 series, published under the general title *Functional safety of*
336 *electrical / electronic / programmable electronic safety-related systems*, can be found on the
337 IEC website.

338 The committee has decided that the contents of this document will remain unchanged until the
339 stability date indicated on the IEC website under webstore.iec.ch in the data related to the
340 specific document. At this date, the document will be

- 341 • reconfirmed,
342 • withdrawn,
343 • replaced by a revised edition, or
344 • amended.

345

346

INTRODUCTION

Systems comprised of electrical and/or electronic elements have been used for many years to perform safety functions in most application sectors. Computer-based systems (generically referred to as programmable electronic systems) are being used in all application sectors to perform non-safety functions and, increasingly, to perform safety functions. If computer system technology is to be effectively and safely exploited, it is essential that those responsible for making decisions have sufficient guidance on the safety aspects on which to make these decisions.

This International Standard sets out a generic approach for all safety lifecycle activities for systems comprised of electrical and/or electronic and/or programmable electronic (E/E/PE) elements that are used to perform safety functions. This unified approach has been adopted in order that a rational and consistent technical policy be developed for all electrically-based safety-related systems. A major objective is to facilitate the development of product and application sector international standards based on the IEC 61508 series.

NOTE 1 Examples of product and application sector international standards based on the IEC 61508 series are given in the bibliography (see references [21], [22] and [37]).

In most situations, safety is achieved by a number of systems which rely on many technologies (for example mechanical, hydraulic, pneumatic, electrical, electronic, programmable electronic). Any safety strategy should therefore consider not only all the elements within an individual system (for example sensors, controlling devices and actuators) but also all the safety-related systems making up the total combination of safety-related systems. Therefore, while this International Standard is concerned with E/E/PE safety-related systems, it may also provide a framework within which safety-related systems based on other technologies may be considered.

It is recognized that there is a great variety of applications using E/E/PE safety-related systems in a variety of application sectors and covering a wide range of complexity, hazard and risk potentials. In any particular application, the required safety measures will be dependent on many factors specific to the application. This International Standard, by being generic, will enable such measures to be formulated in future product and application sector international standards and in revisions of those that already exist.

This International Standard

- considers all relevant overall, E/E/PE system and software safety lifecycle phases (for example, from initial concept, through design, implementation, operation and maintenance to decommissioning) when E/E/PE systems are used to perform safety functions;
- has been conceived with a rapidly developing technology in mind; the framework is sufficiently robust and comprehensive to cater for future developments;
- enables product and application sector international standards, dealing with E/E/PE safety-related systems, to be developed; the development of product and application sector international standards, within the framework of this document, should lead to a high level of consistency (for example, of underlying principles, terminology etc.) both within application sectors and across application sectors; this will have both safety and economic benefits;
- provides a method for the development of the safety requirements specification necessary to achieve the required functional safety for E/E/PE safety-related systems;
- adopts a risk-based approach by which the safety integrity requirements can be determined;
- introduces safety integrity levels for specifying the target level of safety integrity for the safety functions to be implemented by the E/E/PE safety-related systems.
- The standard does not specify the safety integrity level requirements for any safety function, nor does it mandate how the safety integrity level is determined. Instead it provides a risk-based conceptual framework and example techniques.
- sets target failure measures for safety functions carried out by E/E/PE safety-related systems, which are linked to the safety integrity levels;

– sets a lower limit on the target failure measures for a safety function carried out by a single E/E/PE safety-related system. For E/E/PE safety-related systems operating in

- a low demand mode of operation, the lower limit is set at an average probability of a dangerous failure on demand of 10^{-5} ;

- a high demand or a continuous mode of operation, the lower limit is set at an average frequency of a dangerous failure of 10^{-9} [h⁻¹];

NOTE 2 A single E/E/PE safety-related system does not necessarily mean a single-channel architecture.

NOTE 3 It can be possible to achieve designs of safety-related systems with lower values for the target safety integrity for non-complex systems, but these limits are considered to represent what can be achieved for relatively complex systems (for example programmable electronic safety-related systems) at the present time.

– sets requirements for the avoidance and control of systematic faults, which are based on experience and judgement from practical experience gained in industry. Even though the probability of occurrence of systematic failures cannot in general be quantified the standard does, however, allow a claim to be made, for a specified safety function, that the target failure measure associated with the safety function can be considered to be achieved if all the requirements in the standard have been met;

– introduces systematic capability which applies to an element with respect to the confidence that its systematic safety integrity meets the requirements of the specified safety integrity level;

– adopts a broad range of principles, techniques and measures to achieve functional safety for E/E/PE safety-related systems, but does not explicitly use the concept of fail safe. However, the concepts of “fail safe” and “inherently safe” principles may be applicable and adoption of such concepts is acceptable providing the requirements of the relevant clauses in the standard are met.

FUNCTIONAL SAFETY OF ELECTRICAL/ELECTRONIC/ PROGRAMMABLE ELECTRONIC SAFETY-RELATED SYSTEMS –

Part 7: Overview of techniques and measures

1 Scope

1.1 This part of IEC 61508 contains an overview of various safety techniques and measures relevant to IEC 61508-2 and IEC 61508-3.

The references should be considered as basic references to methods and tools or as examples, and may not represent the state of the art.

1.2 IEC 61508-1, IEC 61598-2, IEC 61508-3 and IEC 61508-4 are basic safety publications, although this status does not apply in the context of low complexity E/E/PE safety-related systems (see 3.4.3 of IEC 61508-4). This document provides further information to complement these basic safety publications.

1.3 One of the responsibilities of a technical committee is, wherever applicable, to make use of basic safety publications in the preparation of its publications. In this context, the requirements, test methods or test conditions of this basic safety publication will not apply unless specifically referred to or included in the publications prepared by those technical committees.

1.4 Figure 1 shows the overall framework for parts 1 to 7 of IEC 61508 and indicates the role that IEC 61508-7 plays in the achievement of functional safety for E/E/PE safety-related systems.

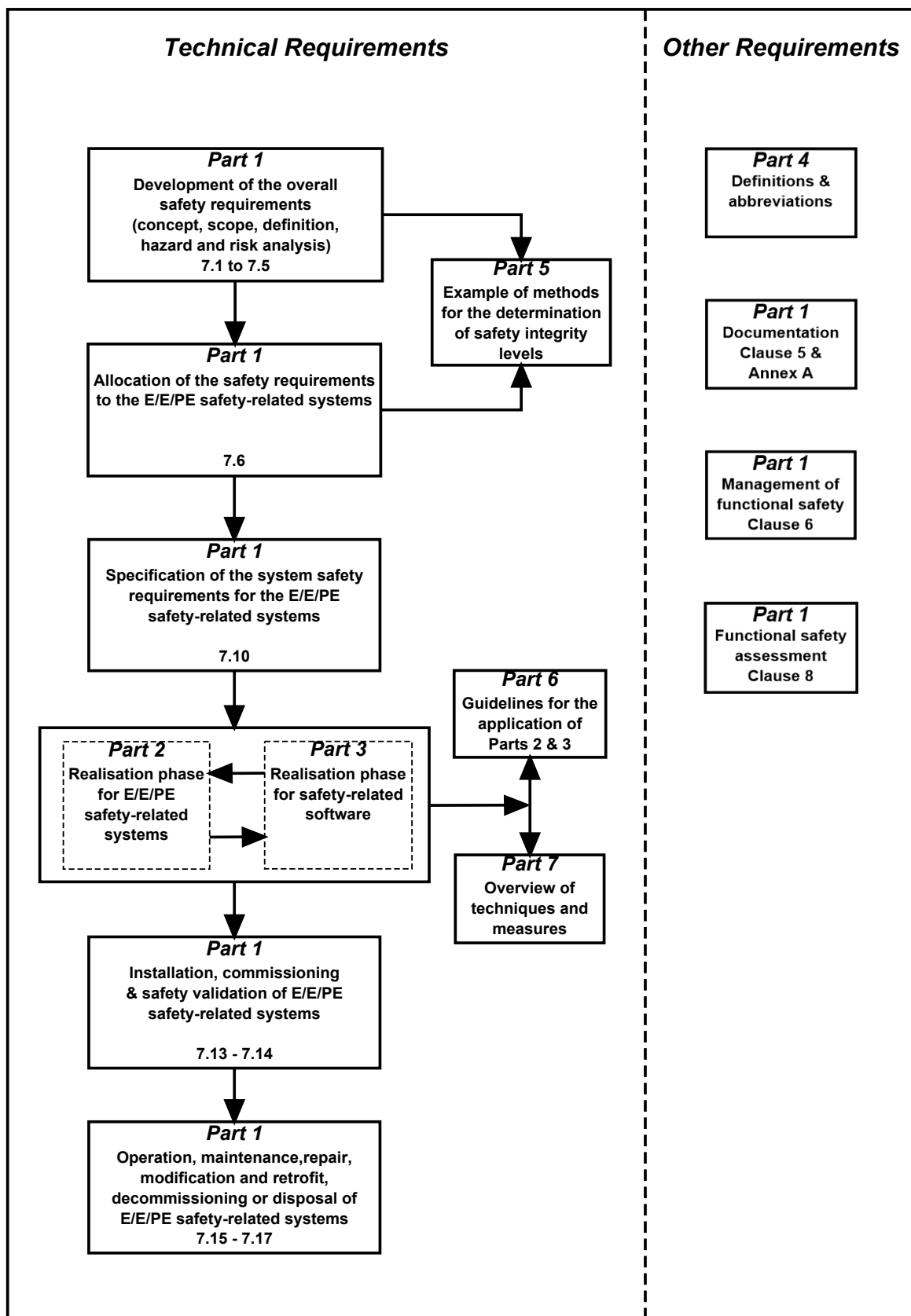


Figure 1 – Overall framework of IEC 61508

2 Normative references

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

IEC 61508-4:202X *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 4: Definitions and abbreviations*

3 Definitions and abbreviations

For the purposes of this document, the definitions and abbreviations given in IEC 61508-4 apply.

Annex A (informative)

Overview of techniques and measures for E/E/PE safety-related systems: control of random hardware failures (see IEC 61508-2)

A.1 Electric

Global objective: To control failures in electromechanical components.

A.1.1 Failure detection by on-line monitoring

NOTE This technique/measure is referenced in Tables A.2, A.3, A.7 and A.13 to A.18 of IEC 61508-2.

Aim: To detect failures by monitoring the behaviour of the E/E/PE safety-related system in response to the normal (on-line) operation of the equipment under control (EUC).

Description: Under certain conditions, failures can be detected using information about (for example) the time behaviour of the EUC. For example, if a switch, which is part of the E/E/PE safety-related system, is normally actuated by the EUC, then if the switch does not change state at the expected time, a failure will have been detected. It is not usually possible to localise the failure.

A.1.2 Monitoring of relay contacts

NOTE This technique/measure is referenced in Tables A.2 and A.14 of IEC 61508-2.

Aim: To detect failures (for example welding) of relay contacts.

Description: Forced contact (or positively guided contact) relays are designed so that their contacts are rigidly linked together. Assuming there are two sets of changeover contacts, *a* and *b*, if the normally open contact, *a*, welds, the normally closed contact, *b*, cannot close when the relay coil is next de-energised. Therefore, the monitoring of the closure of the normally closed contact *b* when the relay coil is de-energised may be used to prove that the normally open contact *a* has opened. Failure of normally closed contact *b* to close indicates a failure of contact *a*, so the monitoring circuit should ensure a safe shut-down, or ensure that shut-down is continued, for any machinery controlled by contact *a*.

References:

Zusammenstellung und Bewertung elektromechanischer Sicherheitsschaltungen für Verriegelungseinrichtungen. F. Kreutzkamp, W. Hertel, Sicherheitstechnisches Informations- und Arbeitsblatt 330212, BIA-Handbuch. 17. Lfg. X/91, Erich Schmidt Verlag, Bielefeld.
www.BGIA-HANDBUCHdigital.de/330212

A.1.3 Comparator

NOTE This technique/measure is referenced in Tables A.2, A.3, A.4 of IEC 61508-2.

Aim: To detect, as early as possible, (non-simultaneous) failures in an independent processing unit or in the comparator.

Description: The signals of independent processing units are compared cyclically or continuously by a hardware comparator. The comparator may itself be externally tested, or it may use self-monitoring technology. Detected differences in the behaviour of the processors lead to a failure message.

A.1.4 Majority voter

NOTE This technique/measure is referenced in Tables A.2, A.3 and A.4 of IEC 61508-2.

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

Description: A voting unit using the majority principle (2 out of 3, 3 out of 3, or m out of n) is used to detect and mask failures. The voter may itself be externally tested, or it may use self-monitoring technology.

Reference:

Guidelines for Safe Automation of Chemical Processes. CCPS, AIChE, New York, 1993, ISBN-10: 0-8169-0554-1, ISBN-13: 978-0-8169-0554-6

A.1.5 Idle current principle (de-energised to trip)

NOTE This technique/measure is referenced in Table A.16 of IEC 61508-2.

Aim: To execute the safety function if power is cut or lost.

Description: The safety function is executed if the contacts are open and no current flows. For example, if brakes are used to stop a dangerous movement of a motor, the brakes are opened by closing contacts in the safety-related system and are closed by opening the contacts in the safety-related system.

Reference:

Guidelines for Safe Automation of Chemical Processes. CCPS, AIChE, New York, 1993, ISBN-10: 0-8169-0554-1, ISBN-13: 978-0-8169-0554-6

A.2 Electronic

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

A.2.1 Tests by redundant hardware

NOTE This technique/measure is referenced in Tables A.3, A.15, A.16 and A.18 of IEC 61508-2.

Aim: To detect failures using hardware redundancy, i.e. using additional hardware not required to implement the process functions.

Description: Redundant hardware can be used to test at an appropriate frequency the specified safety functions. This approach is normally necessary for realising A.1.1 or A.2.2.

A.2.2 Dynamic principles

NOTE This technique/measure is referenced in Table A.3 of IEC 61508-2.

Aim: To detect static failures by dynamic signal processing.

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

Reference:

Elektronik in der Sicherheitstechnik. H. Jürs, D. Reinert, Sicherheitstechnisches Informations- und Arbeitsblatt 330220, BIA-Handbuch, Erich-Schmidt Verlag, Bielefeld, 1993.
<http://www.bgia-handbuchdigital.de/330220>

A.2.3 Standard test access port and boundary-scan architecture

NOTE This technique/measure is referenced in Tables A.3, A.15 and A.18 of IEC 61508-2.

Aim: To control and observe what happens at each pin of an IC.

Description: Boundary-scan test is an IC design technique which increases the testability of the IC by resolving the problem of how to gain access to the circuit test points within it. In a typical boundary-scan IC, comprised of core logic and input and output buffers, a shift-register stage is placed between the core logic and the input and output buffers adjacent to each IC pin. Each shift-register stage is contained in a boundary-scan cell. The boundary-scan cell can control and observe what happens at each input and output pin of an IC, via the standard test access port. Internal testing of the IC core logic is accomplished by isolating the on-chip core logic from stimuli received from surrounding components, and then performing an internal self-test. These tests can be used to detect failures in the IC.

Reference:

IEEE 1149-1:2013, IEEE standard test access port and boundary-scan architecture, IEEE 1011 Computer Society, 2013, ISBN (PDF) 978-0-7381-8263-6 ISBN (Print) 978-0-7381-8264-3 ISBN 978-0-7381-8263

A.2.4 (Not used)**A.2.5 Monitored redundancy**

NOTE This technique/measure is referenced in Table A.3 of IEC 61508-2.

Aim: To detect failure, by providing several functional units, by monitoring the behaviour of each of these to detect failures, and by initiating a transition to a safe condition if any discrepancy in behaviour is detected.

Description: The safety function is executed by at least two hardware channels. The outputs of these channels are monitored and a safe condition is initiated if a fault is detected (i.e. if the output signals from all channels are not identical).

References:

Elektronik in der Sicherheitstechnik. H. Jürs, D. Reinert, Sicherheitstechnisches Informations- und Arbeitsblatt 330220, BIA-Handbuch, Erich-Schmidt Verlag, Bielefeld, 1993.
<http://www.bgia-handbuchdigital.de/330220>

Dependability of Critical Computer Systems 1. F. J. Redmill, Elsevier Applied Science, 1988, ISBN 1-85166-203-0

A.2.6 Electrical/electronic components with automatic check

NOTE This technique/measure is referenced in Table A.3 of IEC 61508-2.

Aim: To detect faults by periodic checking of the safety functions.

Description: The hardware is tested before starting the process, and is tested repeatedly at suitable intervals. The EUC continues to operate only if each test is successful.

References:

Elektronik in der Sicherheitstechnik. H. Jürs, D. Reinert, Sicherheitstechnisches Informations- und Arbeitsblatt 330220, BIA-Handbuch, Erich-Schmidt Verlag, Bielefeld, 1993.
<http://www.bgia-handbuchdigital.de/330220>

Dependability of Critical Computer Systems 1. F. J. Redmill, Elsevier Applied Science, 1988, ISBN 1-85166-203-0

A.2.7 Analogue signal monitoring

NOTE This technique/measure is referenced in Tables A.3 and A.13 of IEC 61508-2.

Aim: To improve confidence in measured signals.

Description: Wherever there is a choice, analogue signals are used in preference to digital on/off states. For example, trip or safe states are represented by analogue signal levels, usually with signal level tolerance monitoring. The technique provides continuity monitoring and a higher level of confidence in the transmitter, reducing the necessary proof-test frequency of the transmitter sensing function. External interfaces, for example impulse lines, will also require testing.

A.2.8 De-rating

NOTE This technique/measure is referenced in 7.4.2.13 of IEC 61508-2.

Aim: To increase the reliability of hardware components.

Description: Hardware components are operated at levels which are guaranteed by the design of the system to be well below the maximum specification ratings. De-rating is the practice of ensuring that under all normal operating circumstances, components are operated well below their maximum stress levels.

A.3 Processing units

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

A.3.1 Self-test by software: limited number of patterns (one-channel)

NOTE This technique/measure is referenced in Table A.4 of IEC 61508-2.

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

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

A.3.2 Self-test by software: walking bit (one-channel)

NOTE This technique/measure is referenced in Table A.4 of IEC 61508-2.

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

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

A.3.3 Self-test supported by hardware (one-channel)

NOTE This technique/measure is referenced in Table A.4 of IEC 61508-2.

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

Description: Additional special hardware facilities support self-test functions to detect failure. For example, this could be a hardware unit which cyclically monitors the output of a certain bit pattern according to the watch-dog principle.

A.3.4 Coded processing (one-channel)

NOTE This technique/measure is referenced in Table A.4 of IEC 61508-2.

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

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

References:

Le processeur codé: un nouveau concept appliqué à la sécurité des systèmes de transports. Gabriel, Martin, Wartski, Revue Générale des chemins de fer, No. 6, June 1990

Vital Coded Microprocessor Principles and Application for Various Transit Systems. P. Forin, IFAC Control Computers Communications in Transportation, 79-84, 1989

A.3.5 Reciprocal comparison by software

NOTE This technique/measure is referenced in Table A.4 of IEC 61508-2.

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

Description: Two processing units exchange data (including results, intermediate results and test data) reciprocally. A comparison of the data is carried out using software in each unit and detected differences lead to a failure message.

A.4 Invariable memory ranges

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

A.4.1 Word-saving multi-bit redundancy (for example ROM monitoring with a modified Hamming code)

NOTE 1 This technique/measure is referenced in Table A.5 of IEC 61508-2.

NOTE 2 See also A.5.6 "RAM monitoring with a modified Hamming code, or detection of data failures with error-detection-correction codes (EDC)" and C.3.2 "Error detecting and correcting codes".

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

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

649 addressing failures, by calculating the redundant bits for the concatenation of the data word
650 and its address.

651 **References:**

652 *Prüfbare und korrigierbare Codes*. W. W. Peterson, München, Oldenburg, 1967

653 *Error detecting and error correcting codes*. R. W. Hamming, The Bell System Technical
654 Journal 29 (2), 147-160, 1950

655 **A.4.2 Modified checksum**

656 NOTE This technique/measure is referenced in Table A.5 of IEC 61508-2.

657 **Aim:** To detect all odd-bit failures, i.e. approximately 50 % of all possible bit failures.

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

664 **A.4.3 Signature of one word (8-bit)**

665 NOTE This technique/measure is referenced in Table A.5 of IEC 61508-2.

666 **Aim:** To detect all one-bit failures and all multi-bit failures within a word, as well as
667 approximately 99,6 % of all possible bit failures.

668 **Description:** The contents of a memory block is compressed (using either hardware or software)
669 using a cyclic redundancy check (CRC) algorithm into one memory word. A typical CRC
670 algorithm treats the whole contents of the block as byte-serial or bit-serial data flow, on which
671 a continued polynomial division is carried out using a polynomial generator. The remainder of
672 the division represents the compressed memory contents – it is the "signature" of the memory
673 – and is stored. The signature is computed once again in later tests and compared with one
674 already stored. A failure message is produced if there is a difference.

675 **A.4.4 Signature of a double word (16-bit)**

676 NOTE This technique/measure is referenced in Table A.5 of IEC 61508-2.

677 **Aim:** To detect all one-bit failures and all multi-bit failures within a word, as well as
678 approximately 99,998 % of all possible bit failures.

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

683 **A.4.5 Block replication (for example double ROM with hardware or 684 software comparison)**

685 NOTE This technique/measure is referenced in Table A.5 of IEC 61508-2.

686 **Aim:** To detect all bit failures.

687 **Description:** The address space is duplicated in two memories. The first memory is operated
688 in the normal manner. The second memory contains the same information and is accessed in
689 parallel to the first. The outputs are compared and a failure message is produced if a difference

is detected. In order to detect certain kinds of bit errors, the data should be stored inversely in one of the two memories and inverted once again when read.

A.5 Variable memory ranges

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

A.5.1 RAM test "checkerboard"

NOTE This technique/measure is referenced in Table A.6 of IEC 61508-2.

Aim: To detect predominantly static bit failures.

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

A.5.2 RAM test "walkpath"

NOTE This technique/measure is referenced in Table A.6 of IEC 61508-2.

Aim: To detect static and dynamic bit failures and cross-talk between memory cells.

Description: The memory range to be tested is initialised by a uniform bit stream. The first cell is then inverted and the remaining memory area is inspected to ensure that the background is correct. After this, the first cell is re-inverted to return it to its original value, and the whole procedure is repeated for the next cell. A second run of the "wandering bit model" is carried out with an inverse background pre-assignment. A failure message is produced if a difference occurs.

A.5.3 RAM test "galpat" or "transparent galpat"

NOTE This technique/measure is referenced in Table A.6 of IEC 61508-2.

Aim: To detect static bit failures and a large proportion of dynamic couplings.

Description: In the RAM test "galpat", the chosen range of memory is first initialised uniformly (i.e. all 0 s or all 1 s). The first memory cell to be tested is then inverted and all the remaining cells are inspected to ensure that their contents are correct. After every read access to one of the remaining cells, the inverted cell is also checked. This procedure is repeated for each cell in the chosen memory range. A second run is carried out with the opposite initialisation. Any difference produces a failure message.

The "transparent galpat" test is a variation on the above procedure: instead of initialising all cells in the chosen memory range, the existing contents are left unchanged and signatures are used to compare the contents of sets of cells. The first cell to be tested in the chosen range is selected, and the signature S1 of all remaining cells in the range is calculated and stored. The cell to be tested is then inverted and the signature S2 of all the remaining cells is recalculated. (After every read access to one of the remaining cells, the inverted cell is also checked.) S2 is compared with S1, and any difference produces a failure message. The cell under test is re-inverted to re-establish the original contents, and the signature S3 of all the remaining cells is recalculated and compared with S1. Any difference produces a failure message. All memory cells in the chosen range are tested in the same manner.

A.5.4 RAM test "Abraham"

NOTE This technique/measure is referenced in Table A.6 of IEC 61508-2.

Aim: To detect all stuck-at and coupling failures between memory cells.

Description: The proportion of faults detected exceeds that of the RAM test "galpat". The number of operations required to perform the entire memory test is about $30n$, where n is the number of cells in the memory. The test can be made transparent for use during the operating cycle by partitioning the memory and testing each partition in different time segments.

Reference:

Efficient Algorithms for Testing Semiconductor Random-Access Memories. R. Nair, S. M. Thatte, J. A. Abraham, IEEE Trans. Comput. C-27 (6), 572-576, 1978

A.5.5 One-bit redundancy (for example RAM monitoring with a parity bit)

NOTE This technique/measure is referenced in Table A.6 of IEC 61508-2.

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

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

A.5.6 RAM monitoring with a modified Hamming code, or detection of data failures with error-detection-correction codes (EDC)

NOTE 1 This technique/measure is referenced in Table A.6 of IEC 61508-2.

NOTE 2 See also A.4.1 "Word-saving multi-bit redundancy (for example ROM monitoring with a modified Hamming code)" and C.3.2 "Error detecting and correcting codes".

Aim: To detect all odd-bit failures, all two-bit failures, some three-bit and some multi-bit failures.

Description: Every access to memory is extended by several redundant bits to produce a modified Hamming code with a Hamming distance of at least 4. Every time data is read, one can determine whether a corruption has taken place by checking the redundant bits. If a difference is found, a failure message is produced. The procedure can also be used to detect addressing failure, when the redundant bits are calculated for the concatenation of the data and its address.

References:

Prüfbare und korrigierbare Codes. W. W. Peterson, München, Oldenburg, 1967

Error detecting and error correcting codes. R. W. Hamming, The Bell System Technical Journal 29 (2), 147-160, 1950

A.5.7 Double RAM with hardware or software comparison and read/write test

NOTE This technique/measure is referenced in Table A.6 of IEC 61508-2.

Aim: To detect all bit failures.

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

A.5.8 RAM test "march"

NOTE This technique/measure is referenced in Table A.6 of IEC 61508-2.

Aim: To detect predominantly persistent bit failures, bit transition failures, addressing failures and linked cell failures.

Description: A pattern of 0s and 1s is written into the cells of memory in a specific pattern and verified in a specific order.

A march test consists of a sequence of elements; each element is a sequence of operations applied to every cell in the memory array before proceeding to the next cell. For example, an operation can consist of writing a 0 into a cell, writing a 1 into a cell, reading an expected 0 from a cell, and reading an expected 1 from a cell. A failure is detected if the expected "1" is not read. The coverage level for linked cells depends on the write/read order.

There are many different types of march tests which are useful for detecting different failure modes. Examples of different march tests and analysis of effectiveness can be found in [47] and [48].

A.6 I/O-units and interfaces (external communication)

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

A.6.1 Test pattern

NOTE This technique/measure is referenced in Tables A.7, A.13 and A.14 of IEC 61508-2.

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

Description: This is a dataflow-independent cyclical test of input and output units. It uses a defined test pattern to compare observations with the corresponding expected values. The test pattern information, the test pattern reception, and test pattern evaluation should all be independent of each other. The EUC should not be inadmissibly influenced by the test pattern.

A.6.2 Code protection

NOTE This technique/measure is referenced in Tables A.7, A.15, A.16 and A.18 of IEC 61508-2.

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

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

A.6.3 Multi-channel parallel output

NOTE This technique/measure is referenced in Table A.7 of IEC 61508-2.

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

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

A.6.4 Monitored outputs

NOTE This technique/measure is referenced in Table A.7 of IEC 61508-2.

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

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

A.6.5 Input comparison/voting

NOTE This technique/measure is referenced in Tables A.7 and A.13 of IEC 61508-2.

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

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

A.7 Data paths (internal communication)

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

A.7.1 One-bit hardware redundancy

NOTE This technique/measure is referenced in Table A.8 of IEC 61508-2.

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

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

A.7.2 Multi-bit hardware redundancy

NOTE This technique/measure is referenced in Table A.8 of IEC 61508-2.

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

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

A.7.3 Complete hardware redundancy

NOTE This technique/measure is referenced in Table A.8 of IEC 61508-2.

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

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

A.7.4 Inspection using test patterns

NOTE This technique/measure is referenced in Table A.8 of IEC 61508-2.

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

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

The test pattern information, the test pattern reception, and test pattern evaluation should all be independent of each other. The EUC should not be inadmissibly influenced by the test pattern.

A.7.5 Transmission redundancy

NOTE This technique/measure is referenced in Table A.8 of IEC 61508-2.

Aim: To detect transient failures in bus communication.

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

A.7.6 Information redundancy

NOTE This technique/measure is referenced in Table A.8 of IEC 61508-2.

Aim: To detect failures in bus communication.

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

A.8 Power supply

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

A.8.1 Overvoltage protection with safety shut-off

NOTE This technique/measure is referenced in Table A.9 of IEC 61508-2.

Aim: To protect the safety-related system against overvoltage.

Description: Overvoltage is detected early enough that all outputs can be switched to a safe condition by the power-down routine or there is a switch-over to a second power unit.

Reference:

Guidelines for Safe Automation of Chemical Processes. CCPS, AIChE, New York, 1993, ISBN-10: 0-8169-0554-1, ISBN-13: 978-0-8169-0554-6

A.8.2 Voltage control (secondary)

NOTE This technique/measure is referenced in Table A.9 of IEC 61508-2.

Aim: To monitor the secondary voltages and initiate a safe condition if the voltage is not in its specified range.

Description: The secondary voltage is monitored and a power-down is initiated, or there is a switch-over to a second power unit, if it is not in its specified range.

A.8.3 Power-down with safety shut-off

NOTE This technique/measure is referenced in Table A.9 of IEC 61508-2.

Aim: To shut off the power with all safety critical information stored.

Description: Overvoltage or undervoltage is detected early enough so that the internal state can be saved in non-volatile memory (if necessary), and so that all outputs can be set to a safe condition by the power-down routine, or that all outputs can be switched to a safe condition by the power-down routine, or there is a switch-over to a second power unit.

A.9 Temporal and logical program sequence monitoring

NOTE This group of techniques and measures is referenced in Tables A.15, A.16 and A.18 of IEC 61508-2.

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

A.9.1 Watch-dog with separate time base without time-window

NOTE This technique/measure is referenced in Tables A.10 and A.11 of IEC 61508-2.

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

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

A.9.2 Watch-dog with separate time base and time-window

NOTE This technique/measure is referenced in Tables A.10 and A.11 of IEC 61508-2.

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

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

A.9.3 Logical monitoring of program sequence

NOTE This technique/measure is referenced in Tables A.10 and A.11 of IEC 61508-2.

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

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

A.9.4 Combination of temporal and logical monitoring of program sequences

NOTE This technique/measure is referenced in Tables A.10 and A.11 of IEC 61508-2.

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

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

A.9.5 Temporal monitoring with on-line check

NOTE This technique/measure is referenced in Tables A.10 and A.11 of IEC 61508-2.

Aim: To detect faults in the temporal monitoring.

Description: The temporal monitoring is checked at start-up, and a start is only possible if the temporal monitoring operates correctly. For example, a heat sensor could be checked by a heated resistor at start-up.

A.10 Ventilation and heating

NOTE This group of techniques and measures is referenced in Tables A.16 and A.18 of IEC 61508-2.

Global objective: To control failures in the ventilation or heating, and/or their monitoring, if this is safety-related.

A.10.1 Temperature sensor

Aim: To detect over- or under-temperature before the system begins to operate outside specification.

Description: A temperature sensor monitors temperature at the most critical points of the E/E/PE safety-related system. Before the temperature leaves the specified range, emergency action is taken.

A.10.2 Fan control

Aim: To detect incorrect operation of the fans.

Description: The fans are monitored for correct operation. If a fan is not working properly, maintenance (or ultimately, emergency) action is taken.

A.10.3 Actuation of the safety shut-off via thermal fuse

Aim: To shut off the safety-related system before the system works outside of its thermal specification.

Description: A thermal fuse is used to shut off the safety-related system. For a PES, the shut-off is introduced by a power-down routine which stores all information necessary for emergency action.

A.10.4 Staggered message from thermo-sensors and conditional alarm

Aim: To indicate that the safety-related system is working outside its thermal specification.

Description: The temperature is monitored and an alarm is raised if the temperature is outside of a specified range.

A.10.5 Connection of forced-air cooling and status indication

Aim: To prevent overheating by forced-air cooling.

Description: The temperature is monitored and forced-air cooling is introduced if the temperature is higher than a specified limit. The user is informed of the status.

A.11 Communication and mass-storage

Global objective: To control failures during communication with external sources and mass-storage.

A.11.1 Separation of electrical energy lines from information lines

NOTE This technique/measure is referenced in Table A.16 of IEC 61508-2.

Aim: To minimise cross-talk induced by high currents in the information lines.

Description: Electrical energy supply lines are separated from the lines carrying the information. The electrical field which could induce voltage spikes on the information lines decreases with distance.

A.11.2 Spatial separation of multiple lines

NOTE This technique/measure is referenced in Tables A.16 of IEC 61508-2.

Aim: To minimise cross-talk induced by high currents in multiple lines.

Description: Lines carrying duplicated signals are separated from each other. The electrical field which could induce voltage spikes on the multiple lines decreases with the distance. This measure also reduces common cause failures.

A.11.3 Design for immunity to electromagnetic interference

NOTE 1 This technique/measure is referenced by clause 7.2.3.3 and in Tables A.16 and A.18 of IEC 61508-2.

Aim: To minimise electromagnetic interference on the safety-related system.

Description: Design techniques such as shielding and filtering are used to increase the interference immunity of the safety-related system to electromagnetic disturbances which may be radiated or conducted on power or signal lines, or result from electrostatic discharges.

NOTE 2 See [16], [17] and [46] for immunity requirements for safety-related systems and for equipment intended to perform safety-related functions (functional safety) in industrial applications.

NOTE 3 Immunity requirement standards like IEC 61000-6-2 cannot reflect the electromagnetic environment in which the safety-related equipment or system is intended to be used. In that case the design techniques to increase electromagnetic interference immunity and immunity testing can consider the immunity necessary for the intended target environment.

References:

IEC 61000-1-2, *Electromagnetic compatibility (EMC) – Part 1-2: General – Methodology for the achievement of functional safety of electrical and electronic systems including equipment with regard to electromagnetic phenomena*

IEC/TR 61000-5-2:1997, *Electromagnetic compatibility (EMC) – Part 5: Installation and mitigation guidelines – Section 2: Earthing and cabling*

994 *Principles and Techniques of Electromagnetic Compatibility*, Second Edition, C. Christopoulos,
995 CRC Press, 2007, ISBN-10: 0849370353, ISBN-13: 978-0849370359

996 *Noise Reduction Techniques in Electronic Systems*. H. W. Ott, John Wiley Interscience,
997 2nd Edition, 1988

998 *EMC for Product Designers*. T. Williams, Newnes, 2007, ISBN 0750681705

999 *Grounding and Shielding Techniques in Instrumentation*, 3rd edition, R. Morrison . Wiley-
1000 Interscience, New York, 1986, ISBN-10: 0471838055, ISBN-13: 978-0471838050

1001 **A.11.4 Antivalent signal transmission**

1002 NOTE This technique/measure is referenced in Tables A.7 and A.16 of IEC 61508-2.

1003 **Aim:** To detect the same induced voltages in multiple signal transmission lines.

1004 **Description:** All duplicated information is transmitted with antivalent signals (for example logic
1005 1 and 0). Common cause failures (for example by electromagnetic emission) can be detected
1006 by an antivalent comparator.

1007 **Reference:**

1008 *Elektronik in der Sicherheitstechnik*. H. Jürs, D. Reinert, Sicherheitstechnisches Informations-
1009 und Arbeitsblatt 330220, BIA-Handbuch. 20. Lfg. V/93, Erich Schmidt Verlag, Bielefeld.
1010 <http://www.bgja-handbuchdigital.de/330220>

1011 **A.12 Sensors**

1012 **Global objective:** To control failures in the sensors of the safety-related system.

1013 **A.12.1 Reference sensor**

1014 NOTE This technique/measure is referenced in Table A.13 of IEC 61508-2.

1015 **Aim:** To detect the incorrect operation of a sensor.

1016 **Description:** An independent reference sensor is used to monitor the operation of a process
1017 sensor. All input signals are checked at suitable time intervals by the reference sensor to detect
1018 failures of the process sensor.

1019 **Reference:**

1020 *Guidelines for Safe Automation of Chemical Processes*. CCPS, AIChE, New York, 1993,
1021 ISBN-10: 0-8169-0554-1, ISBN-13: 978-0-8169-0554-6

1022 **A.12.2 Positive-activated switch**

1023 NOTE This technique/measure is referenced in Table A.13 of IEC 61508-2.

1024 **Aim:** To open a contact by a direct mechanical connection between switch cam and contact.

1025 **Description:** A positive-activated switch opens its normally closed contacts by a direct
1026 mechanical connection between switch cam and contact. This ensures that whenever the switch
1027 cam is in the operated position, the switch contacts should be open.

1028 **Reference:**

1029 *Verriegelung beweglicher Schutzeinrichtungen.* F. Kreutzkamp, K. Becker,
1030 Sicherheitstechnisches Informations- und Arbeitsblatt 330210, BIA-Handbuch. 1. Lfg. IX/85,
1031 Erich Schmidt Verlag, Bielefeld

1032 **A.13 Final elements (actuators)**

1033 **Global objective:** To control failures in the final elements in the safety-related system.

1034 **A.13.1 Monitoring**

1035 NOTE This technique/measure is referenced in Table A.14 of IEC 61508-2.

1036 **Aim:** To detect the incorrect operation of an actuator.

1037 **Description:** The operation of the actuator is monitored (for example by the positively activated
1038 contacts of a relay, see monitoring of relay contacts in A.1.2). The redundancy introduced by
1039 this monitoring can be used to trigger emergency action.

1040 **References:**

1041 *Guidelines for Safe Automation of Chemical Processes.* CCPS, AIChE, New York, 1993, ISBN-
1042 10: 0-8169-0554-1, ISBN-13: 978-0-8169-0554-6

1043 *Zusammenstellung und Bewertung elektromechanischer Sicherheitsschaltungen für Ver-*
1044 *riegelungseinrichtungen.* F. Kreutzkamp, W. Hertel, Sicherheitstechnisches Informations- und
1045 Arbeitsblatt 330212, BIA-Handbuch. 17. Lfg. X/91, Erich Schmidt Verlag, Bielefeld

1046 **A.13.2 Cross-monitoring of multiple actuators**

1047 NOTE This technique/measure is referenced in Table A.14 of IEC 61508-2.

1048 **Aim:** To detect faults in actuators by comparing the results.

1049 **Description:** Each multiple actuator is monitored by a different hardware channel. If a
1050 discrepancy occurs, emergency action is taken.

1051 **A.14 Measures against the physical environment**

1052 NOTE This technique/measure is referenced in Tables A.16 and A.18 of IEC 61508-2

1053 **Aim:** To prevent influences of the physical environment (water, dust, corrosive substances)
1054 causing failures.

1055 **Description:** The enclosure of the equipment is designed to withstand the expected
1056 environment.

1057 **Reference:**

1058 IEC 60529:1989, *Degrees of protection provided by enclosures (IP Code)*

Annex B (informative)

Overview of techniques and measures for E/E/PE safety related systems: avoidance of systematic failures (see IEC 61508-2 and IEC 61508-3)

NOTE Many techniques in this annex can be applicable to software but have not been duplicated in Annex C.

B.1 General measures and techniques

B.1.1 Project management

NOTE This technique/measure is referenced in Tables B.1 to B.6 of IEC 61508-2.

Aim: To avoid failures by adoption of an organisational model and rules and measures for development and testing of safety-related systems.

Description: The most important and best measures are

- the creation of an organisational model, especially for quality assurance which is set down in a quality assurance handbook; and
- the establishment of regulations and measures for the creation and validation of safety-related systems in cross-project and project-specific guidelines.

A number of important basic principles are set down in the following:

- definition of a design organisation:
 - – tasks and responsibilities of the organisational units,
 - – authority of the quality assurance departments,
 - – independence of quality assurance (internal inspection) from development;
- definition of a sequence plan (activity models):
 - – determination of all activities which are relevant during execution of the project including internal inspections and their scheduling,
 - project update;
- definition of a standardised sequence for an internal inspection:
 - planning, execution and checking of the inspection (inspection theory),
 - releasing mechanisms for subproducts,
 - the safekeeping of repeat inspections;
- configuration management:
 - administration and checking of versions,
 - detection of the effects of modifications,
 - consistency inspections after modifications;
- introduction of a quantitative assessment of quality assurance measures:
 - requirement acquisition,
 - failure statistics;
- introduction of computer-aided universal methods, tools and training of personnel.

References:

ISO 9001:2008, *Quality management systems – Requirements*

1099 ISO/IEC 15504 (all parts), *Information technology – Process assessment*

1100 *CMMI: Guidelines for Process Integration and Product Improvement*, 2nd Edition. M. B. Chrissis,
1101 M. Konrad, S. Shrum, Addison-Wesley Professional, 2006, ISBN-10: 0-3212-7967-0, ISBN-13:
1102 978-0-3212-7967-5

1103 *Guidelines for Safe Automation of Chemical Processes*. CCPS, AIChE, New York, 1993, ISBN-
1104 10: 0-8169-0554-1, ISBN-13: 978-0-8169-0554-6

1105 *Dependability of Critical Computer Systems* 1. F. J. Redmill, Elsevier Applied Science, 1988,
1106 ISBN 1-85166-203-0

1107 **B.1.2 Documentation**

1108 NOTE 1 This technique/measure is referenced in Tables B.1 to B.6 of IEC 61508-2.

1109 NOTE 2 See also Clause 5 and Annex A of IEC 61508-1.

1110 **Aim:** To avoid failures and facilitate assessment of system safety, by documenting each step
1111 during development.

1112 **Description:** The operational capacity and safety, as well as the care taken in development by
1113 all parties involved, has to be demonstrated during assessment. In order to be able to show the
1114 development care, and in order to guarantee the verification of the evidence of safety at any
1115 time, special importance is given to the documentation.

1116 Important common measures are the introduction of guidelines and computer aid, i.e.

- 1117 – guidelines, which
 - 1118 • specify a grouping plan;
 - 1119 • ask for checklists for the contents; and
 - 1120 • determine the form of the document;
- 1121 – administration of the documentation with the help of a computer-aided and organised project
1122 library.

1123 Individual measures are:

- 1124 – separation in the documentation
 - 1125 • of the requirements,
 - 1126 • of the system (user-documentation) and
 - 1127 • of the development (including internal inspection);
- 1128 – grouping of the development documentation according to the safety lifecycle;
- 1129 – definition of standardised documentation modules, from which the documents can be
1130 compiled;
- 1131 – clear identification of the constituent parts of the documentation;
- 1132 – formalised revision update;
- 1133 – selection of clear and intelligible means of description:
 - 1134 • formal notation for determinations;
 - 1135 • natural language for introductions, justifications and representations of intentions;
 - 1136 • graphical representations for examples;
 - 1137 • semantic definition of graphical elements; and
 - 1138 • directories of specialised words.

References:

IEC 61506:1997, *Industrial-process measurement and control – Documentation of application software*

Guidelines for Safe Automation of Chemical Processes. CCPS, AIChE, New York, 1993, ISBN-10: 0-8169-0554-1, ISBN-13: 978-0-8169-0554-6

B.1.3 Separation of E/E/PE system safety functions from non-safety functions

NOTE This technique/measure is referenced in Tables B.1 and B.6 of IEC 61508-2.

Aim: To prevent the non-safety-related part of the system from influencing the safety-related part in undesired ways.

Description: In the specification, it should be decided whether a separation of the safety-related systems and non-safety-related systems is possible. Clear specifications should be written for the interfacing of the two parts. A clear separation reduces the effort for testing the safety-related systems.

Reference:

Guidelines for Safe Automation of Chemical Processes. CCPS, AIChE, New York, 1993, ISBN-10: 0-8169-0554-1, ISBN-13: 978-0-8169-0554-6

B.1.4 Diverse hardware

NOTE This technique/measure is referenced in Tables A.15, A.16 and A.18 of IEC 61508-2.

Aim: To detect systematic failures during operation of the EUC, using diverse components with different rates and types of failures.

Description: Different types of components are used for the diverse channels of a safety-related system. This reduces the probability of common cause failures (for example overvoltage, electromagnetic interference), and increases the probability of detecting such failures.

Existence of different means of performing a required function, for example different physical principles, offer other ways of solving the same problem. There are several types of diversity. Functional diversity employs the use of different approaches to achieve the same result.

Reference:

Guidelines for Safe Automation of Chemical Processes. CCPS, AIChE, New York, 1993, ISBN-10: 0-8169-0554-1, ISBN-13: 978-0-8169-0554-6

B.1.5 Traceability

Aim: To maintain consistency within and between lifecycle stages.

Description: In order to ensure that the results from lifecycle activities meet the aim of developing a correctly operating safety-related system, it is essential to ensure consistency, completeness and correctness between inputs and outputs to and from the lifecycle stages. A key concept here is that of “traceability” between activities. This is essentially an impact analysis to check

- that decisions made at an earlier stage are adequately implemented in later stages (forward traceability), and
- that decisions made at a later stage are actually required and mandated by earlier decisions (backward traceability).

Forward traceability is broadly concerned with checking that a requirement is adequately addressed in later lifecycle stages. For example, an existing requirement signals that a corresponding test case should be available.

Backward traceability is broadly concerned with checking that every implementation (interpreted in a broad context, and not confined to code implementation) decision is clearly justified by some requirement. If this justification is absent, then the implementation contains something unnecessary that will add to the complexity but not necessarily address any genuine requirement of the safety-related system. For example if there is a test case, it should be traceable to a requirement. An existing test case without a requirement can reveal either that a requirement is missing or the test case is not necessary.

Both forms of traceability can be used to trace to the first or the last lifecycle activities by completely tracing back to the requirement or completely tracing forward to the finally implemented solution. Traceability in both directions can improve all of the involved statements on the respective lifecycle stages as a formulated design solution might reveal a weakness in the stated requirement and vice versa.

But when traceability is given from the input into each lifecycle activity to the output this will result in a completely traced path through the lifecycle stages just as well. This document proposes a way that seems adequate for all sizes of project, however traceability that is more granular can be chosen if this adds value like displaying the relation between input and output to this analysis in a more comprehensive way.

Reference:

Requirements Engineering. E. Hull, K. Jackson, J. Dick. Springer, 2005, ISBN 1852338792, 9781852338794

1202

B.1.6 Functional Safety Assurance Role Independence

NOTE 1 This technique/measure is referenced in IEC 61508-1, clauses 6.2.7, 7.8.2, 7.18, 8, 9, and Annex B.

Aim: To uncover undetected systematic failures by adoption of an organizational model and rules and measures for establishing independence between those responsible for specific safety lifecycle activities and those responsible for the functional safety assurance (i.e. verification, validation, assessment, or audit) of those specific activities.

Description: Assurance activities (*i.e. verification, validation, assessment, and audit*) are imposed to substantiate that requirements have been fulfilled, have been fulfilled for a specific intended use, that planned procedures have been complied with and implemented effectively, and that functional safety has been achieved. The specific techniques and measures for implementing these activities may include any combination of:

- examination through review or inspection,
- analysis of key properties or characteristics,
- simulation or formal proof, and
- test.

The preferred method or combination of methods may also be stipulated for a given activity.

This collection of activities and implementation techniques can be generally categorized as functional safety assurance.

The value of establishing independence between lifecycle activity and assurance roles is to ensure that these functional safety assurance activities are not unduly prejudiced, either:

- 1223 • by implicit knowledge and bias that may exist with the author or developer of the output,
1224 or
- 1225 • by an authoritative influencer whose priority and accountability are for something other
1226 than functional safety assurance.

1227 These two concerns can be characterized as a need for “technical independence” and a need
1228 for “management independence”.

1229 **Technical Independence:** This aspect recognizes that an author or developer may have pre-
1230 conceived ideas about the acceptable configurations, functions, or features of an artefact or
1231 output that do not align directly with the phase inputs; rather, they are due to the developer’s
1232 past experience or experience during design or implementation of the artefact. It is important
1233 for those responsible for functional safety assurance activities to be free of this potential
1234 developer bias in order to evaluate the output solely based upon the appropriate inputs (i.e.
1235 requirements, design or implementation guidelines, standards, etc.). This aspect of
1236 independence is valuable and necessary in all assurance circumstances.

1237 NOTE 2 This form of independence can be referred to as “intellectual independence” in some industries.

1238 **Management Independence:** This aspect acknowledges that having a common authority (e.g.
1239 common management) having accountability for both oversight and influence over developer
1240 and functional safety assurance roles, may generate a conflict of priorities for those responsible
1241 for functional safety assurance. Generating formal independence criteria for the reporting
1242 structure, between the developer and assurance roles can ensure that such conflicts will be
1243 addressed at an appropriate level of authority where competing priorities can be given
1244 appropriate weight and consideration.

1245 NOTE 3 This form of independence is referred to as “organisational independence” in some industries.

1246 The following tables are provided to breakdown and illustrate the relationships between
1247 functional safety assurance activities, independence aspects, and independence measures.

1248 **Table B.1 – Significance of Independence Concern vs.**
1249 **Functional Safety Assurance Activity**

Functional Safety Assurance Activities	Significance	
	Technical	Management
Verification	++	-
Validation	++	+
Assessment	+	++
Audit	-	++
Key: - Less significant + Significant ++ More significant		

1250
1251 **Table B.2 – Effectiveness of Independence Measure vs. Independence Concern**

Independence Aspect	Effectiveness of Independence Measure		
	Technical Level 1	Technical & Management Level 2	Technical & Management Level 3
Technical	++	+	+
Management	-	+	++



Key:

- Less effective
- + Effective
- ++ Most effective

Table B.3 – Effectiveness of Independence Measure vs. Functional Safety Assurance Activity

Functional Safety Assurance Activities	Effectiveness of Independence Measure		
	Technical Level 1*	Technical & Management Level 2*	Technical & Management Level 3*
Verification	++	+	-
Validation	+	++	+
Assessment	-	+	++
Audit	-	+	++
Key: - Less effective + Effective ++ Most effective			

The contents of these tables can be used to justify independence requirements for each functional safety assurance activity.

Reference:

Legal Studies, Volume 18, Issue 1, March 1998, pp.78-100
 DOI: <https://doi.org/10.1111/j.1748-121X.1998.tb00070.x>
 Copyright © Society of Legal Scholars 1998

B.2 E/E/PE system design requirements specification

Global objective: To produce a specification which is, as far as possible, complete, free from mistakes, free from contradiction, and simple to verify.

B.2.1 Structured specification

NOTE This technique/measure is referenced in Tables B.1 and B.6 of IEC 61508-2.

Aim: To reduce complexity by creating a hierarchical structure of partial requirements. To avoid interface failures between the requirements.

Description: This technique structures the functional specification into partial requirements such that the simplest possible, visible relations exist between the latter. This analysis is successively refined until small clear partial requirements can be distinguished. The result of the final refinement is a hierarchical structure of partial requirements which provide a framework for the specification of the complete requirements. This method emphasises the interfaces of the partial requirements and is particularly effective for avoiding interface failures.

References:

ESA PSS 05-02, *Guide to the user requirements definition phase*, Issue 1, Revision 1, ESA Board for Software Standardisation and Control (BSSC), ESA, Paris, March 1995,
<ftp://ftp.estec.esa.nl/pub/wm/wme/bssc/PSS0502.pdf>



B.2.2 Formal methods

NOTE 1 This technique/measure is referenced in Tables B.1, B.2 and B.6 of IEC 61508-2.

Aim: Formal methods transfer the principles of mathematical reasoning to the specification and implementation of technical systems therefore increase the completeness, consistency or correctness of a specification or implementation.

Formal methods encompass a large part of the safety lifecycle including specification, design, development and verification. They apply as well to hardware as to software as they describe system behaviours. They require a specific organisation of the engineering that is often implicit with the engineering environment provided with the tools associated with the methods

Description: Formal methods provide a means of developing a description of a system during specification and/or implementation phase. These formal descriptions are mathematical or logical descriptions or models of the system function and/or structure. See the definition of formal methods in IEC 61508-4. The use of formal methods, along with lists of types of methods which can be used to establish exact properties of software and its documentation are to be found in IEC TS 61508-3-2.

Unambiguous system description can be achieved (e.g. any state of an automaton is described by its initial state, inputs and the transition equations of the automaton) which increase understanding of the underlying system.

Choosing a suitable formal method is a difficult undertaking requiring full understanding of system, its development process and the range of mathematical models that could possibly be used (see following notes).

NOTE 2 The theorems of interest of the model (properties) can represent guaranties about the system which provides far more confidence than simulation i.e. observing selected actions of the system.

NOTE 3 The disadvantages of formal methods can be:

- Fixed level of abstraction;
- limitations to capture all functionality that is relevant at the given stage;
- difficulty that implementation engineers have to understand the model;
- high efforts to develop, analyze and maintain model over the lifecycle of system;
- availability of efficient tools which support the building and analysis of model;
- availability of staff capable to develop and analyze model.

NOTE 4 Modeling of the target function of system can often deemphasize the fault robustness of a system. Therefore respective formal methods including system robustness can be selected.

References:

IEC TS 61508-3-2 Functional safety of electrical/electronic/programmable electronic safety-related system – Part 3-2: Requirements and guidance in the use of mathematical and logical techniques for establishing exact properties of software and its documentation

Problem Frames: Analysing and Structuring Software Development, Michael Jackson, ACM Press/Pearson Education, 2001.

Specifying Systems, Leslie Lamport, Addison-Wesley/Pearson Education 2003. A version is available from the author on his WWW pages.

The SPIN Model Checker: Primer and Reference Manual, Gerard J. Holzmann, Addison-Wesley/Pearson Education 2004. Software, literature and guides are to be found on the WWW.

High Integrity Software, John Barnes with Praxis, Addison-Wesley/Pearson Education 2006. A version is available on the WWW

1323 *Software Abstractions: Logic, Language and Analysis, Daniel Jackson, Revised Edition, MIT*
 1324 *Press, 2016. Alloy theorem-proving software is available on the WWW.*

1325 *Using Z; Specification, Refinement and Proof, Jim Woodcock and Jim Davies, Prentice-*
 1326 *Hall/Pearson Education, 1996. A version is available on the WWW.*

1327 *The Z Notation: A Reference Manual, J.M. Spivey, Prentice-Hall, 1989. A version is available*
 1328 *on the WWW.*

1329 *Communicating Sequential Processes, C. A. R. Hoare, Prentice-Hall, 1985. A version is*
 1330 *available on the WWW.*

1331 *Choosing a Formal Method, Formal Methods Europe. Available on-line at*
 1332 *<https://www.fmeurope.org/choosingaformalmethod/>*

1333 **B.2.3 Semi-formal methods**

1334 **Aim:** To express parts of a specification or of a design unambiguously and consistently, so
 1335 that some mistakes, omissions and wrong behaviour can be detected, but not necessarily all of
 1336 them.

1337 NOTE– This technique/measure is referenced in Tables B.1, B.2 and B.6 of IEC 61508-2 and in Tables A.1, A.2, A.4,
 1338 B.7, C.1, C.2, C.4 and C.17 of IEC 61508-3.

1339 **B.2.3.1 General**

1340 **Aim:** To assure that the design meets its specification.

1341 **Description:** Semi-formal methods use notations that have a defined syntax and semantics,
 1342 but not necessarily a formal semantics. Semi-formal methods provide a means of developing a
 1343 description of a system at some stage in its development, i.e. specification, design or coding.
 1344 The description can in some cases be analysed by machine or animated to display various
 1345 aspects of the system behaviour. Animation can give extra confidence that the system meets
 1346 the real requirement as well as the specified requirement. Semi-formal methods may have a
 1347 mathematical or logical basis, for example, finite-state machines, timed Petri nets, and message
 1348 flow diagrams (also sometimes called sequence charts). However, the modelling of a system
 1349 specification in such a mathematical or logical medium may not be rigorous; or the semantics
 1350 of the medium may not be completely or unambiguously defined in all cases. This allows some
 1351 errors to be manifest in development, namely those highlighted by the mathematics of the
 1352 modelling method, but other errors may lie in the inexact modelling, or in ambiguous or
 1353 incomplete semantics of specific constructs, and these errors will not be detected in use of the
 1354 semi-formal technique.

1355 Three modelling techniques that can be used as semi-formal methods are described in the
 1356 following subclauses.

1357 **B.2.3.2 Problem Frames**

1358 **Aim:** To model system functional requirements, to decompose requirements, where possible
 1359 into “subproblems” and to enable partial checking of the coherence of the
 1360 requirements and decomposition, as well as the freedom of the functional
 1361 requirements from contradiction

1362 **Description:** The functional requirements are expressed at a high level as a “problem” to be
 1363 solved of existing entities in (what is to become) the system environment. Through
 1364 specifying and analysing the “problem”, it may be decomposed in given ways into
 1365 “subproblems” that can end up being “solved” through known templates.

1366 **References:** Problem Frames, Michael Jackson, ACM Press/Pearson Education, 2000

B.2.3.3 Finite state machines / state transition diagrams

NOTE This technique/measure is referenced in Tables B.5, C.15 and C.17 of IEC 61508-3.

Aim: To model, verify, specify or implement the control structure of a system.

Description: Many systems can be described in terms of their states, their inputs, and their actions. Thus when in state S1, on receiving input I a system might carry out action A and move to state S2. By describing a system's actions for every input in every state we can describe a system completely. The resulting model of the system is called a finite state machine (or finite state automata). It is often drawn as a so-called state transition diagram showing how the system moves from one state to another, or as a matrix in which the dimensions are state and input, and the matrix cells contain the action and new state resulting from receiving the input when in the given state.

Where a system is complicated or has a natural structure, this can be reflected in a layered finite state machine. A Statechart is a type of state transition diagram in which nested states are allowed (the object state splits into two or more sub-states which can evolve in parallel, and possibly recombine into a single state at some point); this adds to the expressive power of the state transition notation but can add extra complexity which may be undesirable in a safety related system. Statecharts have a formal (mathematical) specification. State transition diagrams can apply to the whole system or to some object or element within it.

A specification or design expressed as a finite state machine can be checked for

1. completeness (the system or object must have an action and new state for every input in every state);
2. consistency (only one state transition is possible for each state/input pair); and
3. reachability (whether or not it is possible to get from one state to another by any sequence of inputs); and
4. absence of endless loops or dead-end states; etc.

These are important properties for critical systems. Tools to support these checks are easily developed and various models based on finite state automata (formal languages, Petri nets, Markov graphs, etc.) can be used. Algorithms also exist that allow the automatic generation of test cases for verifying a finite state machine implementation or for animating a finite state machine model. State transition diagrams and Statecharts are widely supported by tools which allow the diagrams to be drawn and checked, and which will generate code to implement the described state machine.

They can also be used for failure probability calculations, see B.6 and C.6.

References:

Introduction to Automata Theory, Languages, and Computation (3rd Edition). J. Hopcroft, R. Motwani, J. Ullman, Addison-Wesley Longman Publishing Co, 2006, ISBN: 0321462254

Sécurisation des architectures informatiques. Jean-Louis Boulanger, Hermès – Lavoisier, 2009, ISBN: 978-2-7462-1991-5

B.2.3.4 Time Petri nets

NOTE This technique/measure is referenced in Tables B.5, C.15 and C.17 of IEC 61508-3.

Aim: To model relevant aspects of the system behaviour and to assess and possibly improve safety and operational requirements through analysis and re-design.

Description: Petri nets are a particular case of finite state automata. They belong to a class of graph theoretic models which are suitable for representing information and control flow in systems that exhibit concurrency and have asynchronous behaviour.

1412 A Petri net is a network of places and transitions. The places may be "marked" or "unmarked".
1413 A transition is "enabled" when all the input places to it are marked. When enabled, it is permitted
1414 (but not obliged) to "fire". If it fires, the input places to the transition become unmarked, and
1415 each output place from the transition is marked instead.

1416 The potential hazards can be represented as particular states (markings) in the model. The
1417 Petri net model can be extended to allow for timing features of the system. Although "classical"
1418 Petri nets concentrate on control flow aspects, several extensions have been proposed to
1419 incorporate data flow into the model.

1420 They also provide a very efficient support for performing Monte Carlo simulation in order to
1421 achieve failure probability calculations, see B.6.6.8.

1422 **References:**

1423 *Timed Petri Nets: Theory and Application*. Jiacun Wang, Springer, 1998, ISBN 0792382706

1424 *Sécurisation des architectures informatiques*. Jean-Louis Boulanger, Hermès – Lavoisier, 2009,
1425 ISBN: 978-2-7462-1991-5

1426 **B.2.4 Computer-aided specification tools**

1427 NOTE This technique/measure is referenced in Tables B.1 and B.6 of IEC 61508-2 and in Tables A.1, A.2, C.1 and
1428 C.2 of IEC 61508-3.

1429 **B.2.4.1 General**

1430 **Aim:** To use formal specification techniques to facilitate automatic detection of ambiguity and
1431 completeness.

1432 **Description:** The technique produces a specification in the form of a database that can be
1433 automatically inspected to assess consistency and completeness. The specification tool can
1434 animate various aspects of the specified system to the user. In general, the technique supports
1435 not only the creation of the specification but also of the design and other phases of the project
1436 lifecycle. Specification tools can be classified according to the following subclauses.

1437 **B.2.4.2 Tools oriented towards no specific method**

1438 **Aim:** To help the user write a good specification by providing prompts and links between related
1439 parts.

1440 **Description:** The specification tool takes over some routine work from the user and supports
1441 the project management. It does not enforce any particular specification methodology. The
1442 relative independence with regard to method allows users a great deal of freedom but also gives
1443 them little of the specialised support necessary when creating specifications. This makes
1444 familiarisation with the system more difficult.

1445 **B.2.4.3 Model orientated procedure with hierarchical analysis**

1446 **Aim:** To avoid incompleteness, ambiguity and contradiction in the specification, e.g. supporting
1447 the user writing a good specification by ensuring consistency between descriptions of actions
1448 and data at various levels of abstraction.

1449 **Description:** This method gives a functional representation of the desired system (structured
1450 analysis) at various levels of abstraction (degree of precision). There is a huge arsenal of such
1451 models: finite automata are a class of such models widely used to describe the evolution of
1452 discrete/digital systems. Differential equations are similar in spirit and aim at
1453 continuous/analogue systems. The analysis at various levels acts on both actions and data.
1454 Assessment of ambiguity and completeness is possible between hierarchical levels as well as

1455 between two functional units (modules) on the same level (e.g. any state of a system model is
1456 described by its initial state, inputs and the transition equations of the automaton).

1457 NOTE Issues of model based descriptions may be the level of abstraction, limitations to capture all functionality
1458 that is relevant at the given stage, difficulty that practitioners have to understand the model (from reading the syntax
1459 to understanding), high efforts to develop, analyze and maintain a model over the lifecycle of a system, availability
1460 of efficient tools which support the building and analysis of model (development of such tools is certainly a high effort
1461 undertaking) and availability of staff capable to develop and analyze models.

1462 **Reference:**

1463 *System requirements analysis*. Jeffrey O. Grady, Academic Press, 2006, ISBN 012088514X,
1464 9780120885145

1465 **B.2.4.4 Entity-relationship-attribute data models**

1466 **Aim:** To help the user write a good specification by focusing on entities within the system and
1467 relationships between them.

1468 **Description:** The desired system is described as a collection of objects and their relationships.
1469 The tool enables one to determine which relationships can be interpreted by the system. In
1470 general, the relationships permit a description of the hierarchical structure of the objects, the
1471 data flow, the relationships between the data, and which data are subject to certain
1472 manufacturing processes. The classical procedure has been extended for process control
1473 applications. Inspection capabilities and support for the user depend on the variety of
1474 relationships illustrated. On the other hand, a large number of representation possibilities
1475 makes the application of this technique complex.

1476 **Reference:**

1477 *Software Requirements: Practical Techniques for Gathering and Managing Requirements*
1478 *Throughout the Product Development Cycle*. Karl Eugene Wiegers, Microsoft Press, 2003, ISBN
1479 0735618798, 9780735618794

1480 **B.2.4.5 Incentive and answer**

1481 **Aim:** To help the user write a good specification by identifying stimulus-response relationships.

1482 **Description:** The relationships between the objects of the system are specified in a notation of
1483 "incentives" and "answers". A simple and easily expanded language is used which contains
1484 language elements which represent objects, relationships, characteristics and structures.

1485 **B.2.5 Checklists**

1486 NOTE This technique/measure is referenced in Tables B.1, B.2 and B.6 of IEC 61508-2 and in Tables A.10, B.8,
1487 C.10 and C.18 of IEC 61508-3.

1488 **Aim:** To draw attention to, and manage critical appraisal of, all important aspects of the system
1489 by safety lifecycle phase, ensuring comprehensive coverage without laying down exact
1490 requirements.

1491 **Description:** A set of questions to be answered by the person performing the checklist. Many
1492 of the questions are of a general nature and the assessor should interpret them as seems most
1493 appropriate to the particular system being assessed. Checklists can be used for all phases of
1494 the overall, E/E/PE system safety and software safety lifecycles and are particularly useful as
1495 a tool to aid the functional safety assessment.

1496 To accommodate wide variations in systems being validated, most checklists contain questions
1497 which are applicable to many types of system. As a result, there may well be questions in the
1498 checklist being used which are not relevant to the system being dealt with and which should be

1499 ignored. Equally there may be a need, for a particular system, to supplement the standard
1500 checklist with questions specifically directed at the system being dealt with.

1501 In any case it should be clear that the use of checklists depends critically on the expertise and
1502 judgement of the engineer selecting and applying the checklist. As a result, the decisions taken
1503 by the engineer, with regard to the checklist(s) selected, and any additional or superfluous
1504 questions, should be fully documented and justified. The objective is to ensure that the
1505 application of the checklists can be reviewed and that the same results will be achieved unless
1506 different criteria are used.

1507 The object in completing a checklist is to be as concise as possible. When extensive justification
1508 is necessary this should be done by reference to additional documentation. Pass, fail and
1509 inconclusive, or some similar restricted set of responses should be used to document the results
1510 for each question. This conciseness greatly simplifies the procedure of reaching an overall
1511 conclusion as to the results of the checklist assessment.

1512 **References:**

1513 IEC 60880:2006, *Nuclear power plants – Instrumentation and control systems important to*
1514 *safety – Software aspects for computer-based systems performing category A functions*

1515 *The Art of Software Testing, third edition.* G. J. Myers, C. Sandler, T. Badgett, John Wiley and
1516 Sons, 2011, ISBN 978-1-118-13315-6

1517 *Software Testing: A Craftsman's Approach, fourth edition,* P. C. Jorgensen, Auerbach
1518 Publications, 2013, ISBN 978-1-4665-6068-0

1519 *Software Quality Assurance: From Theory to Implementation.* Daniel Galin, Pearson Education,
1520 2004, ISBN 0201709457, 9780201709452

1521 IEC 61346 (all parts), *Industrial systems, installations and equipment and industrial products –*
1522 *Structuring principles and reference designation*

1523 **B.2.5.1 Guidelines for Safe Automation of Chemical Processes. CCPS, AIChE, New**
1524 **York, 1993, ISBN-10: 0-8169-0554-1, ISBN-13: 978-0-8169-0554-6**

1525 *Risk Assessment and Risk Management for the Chemical Process Industry.* H.R. Greenberg,
1526 J.J. Cramer, John Wiley and Sons, 1991, ISBN 0471288829, 9780471288824

1527 **B.2.6 Inspection of the specification**

1528 NOTE This technique/measure is referenced in Tables B.1 and B.6 of IEC 61508-2.

1529 **Aim:** To avoid incompleteness and contradiction in the specification.

1530 **Description:** Inspection is a general technique in which various qualities of a specification
1531 document are assessed by an independent team. The inspection team puts questions to the
1532 creator, who should answer them satisfactorily. The examination should (if possible) be carried
1533 out by a team that was not involved in the creation of the specification. The required degree of
1534 independence is determined by the safety integrity levels demanded of the system. The
1535 independent inspectors should be able to reconstruct the operational function of the system in
1536 an indisputable manner without referring to any further specifications. They should also check
1537 that all relevant safety and technical aspects in the operational and organisational measures
1538 are covered. This procedure has proved itself to be very effective in practice.

1539 **References:**

1540 IEC 61160:2005, *Design review*

1541 *The Art of Software Testing, third edition.* G. J. Myers, C. Sandler, T. Badgett, John Wiley and
1542 Sons, 2011, ISBN 978-1-118-13315-6

1543 *Software Testing: A Craftsman's Approach, fourth edition,* P. C. Jorgensen, Auerbach
1544 Publications, 2013, ISBN 978-1-4665-6068-0

1545 *Software Quality Assurance: From Theory to Implementation.* D. Galin, Pearson Education,
1546 2004, ISBN 0201709457, 9780201709452

1547 **B.3 E/E/PE system design and development**

1548 **Global objective:** To produce a stable design of the safety-related system in conformance with
1549 the specification.

1550 **B.3.1 Observance of guidelines and standards**

1551 NOTE This technique/measure is referenced in Table B.2 of IEC 61508-2.

1552 **Aim:** To observe application sector standards (not specified in this document).

1553 **Description:** Guidelines should be complied with during the design of the safety-related system.
1554 These guidelines should firstly lead to safety-related systems which are practically free from
1555 failures, and secondly facilitate the subsequent safety validation. They can be universally valid,
1556 specific to a project, or specific only to a single phase.

1557 **Reference:**

1558 *Guidelines for Safe Automation of Chemical Processes.* CCPS, AIChE, New York, 1993, ISBN-
1559 10: 0-8169-0554-1, ISBN-13: 978-0-8169-0554-6

1560 **B.3.2 Structured design**

1561 NOTE This technique/measure is referenced in Tables B.2 and B.6 of IEC 61508-2.

1562 **Aim:** To reduce complexity by creating a hierarchical structure of the design to simplify
1563 verification.

1564 **Description:** When designing the hardware, specific criteria or methods should be used. For
1565 example, the following might be required:

- 1566 1. a hierarchically structured circuit design;
1567 2. use of manufactured and tested circuit parts.

1568

1569 **References:**

1570 IEC 61346 (all parts), *Industrial systems, installations and equipment and industrial products –*
1571 *Structuring principles and reference designation*

1572

1573 **B.3.3 Use of well-tried components**

1574 NOTE This technique/measure is referenced in Tables B.2 and B.6 of IEC 61508-2.

1575 **Aim:** To reduce the risk of numerous first time and undetected faults by the use of components
1576 with specific characteristics.

Description: The selection of well-trying components is carried out by the manufacturer, with regard to safety according to the reliability of the elements (for example the use of operationally tested physical units to meet high safety requirements, or the storing of safety-related programs in safe memories only). The safety of memories can refer to unauthorised access as well as environmental influences (electromagnetic compatibility, radiation, etc) and the response of the elements in the event of a failure occurring.

Reference:

IEC 61163-1:2006, *Reliability stress screening – Part 1: Repairable assemblies manufactured in lots*

B.3.4 Modularisation

NOTE This technique/measure is referenced in Tables B.2 and B.6 of IEC 61508-2.

Aim: To reduce complexity and avoid failures, related to interfacing between subsystems.

Description: Every subsystem, at all levels of the design, is clearly defined and is of restricted size (only a few functions). The interfaces between subsystems are kept as simple as possible and the cross-section (i.e. shared data, exchange of information) is minimised. The complexity of individual subsystems is also restricted.

References:

The Art of Software Testing, third edition. G. J. Myers, C. Sandler, T. Badgett, John Wiley and Sons, 2011, ISBN 978-1-118-13315-6

Software Testing: A Craftsman's Approach, fourth edition, P. C. Jorgensen, Auerbach Publications, 2013, ISBN 978-1-4665-6068-0

Software Engineering for Real-time Systems. J. E. Cooling, Pearson Education, 2003, ISBN 0201596202, 9780201596205

Software Reliability – Principles and Practices. G. J. Myers, Wiley-Interscience, New York, 1976, ISBN-10: 0471627658, ISBN-13: 978-0471627654

B.3.5 Computer-aided design tools

NOTE This technique/measure is referenced in Tables B.2 and B.6 of IEC 61508-2 and in Tables A.4 and C.4 of IEC 61508-3.

Aim: To carry out the design procedure more systematically. To include appropriate automatic construction elements which are already available and tested.

Description: Computer-aided design tools (CAD) should be used during the design of both hardware and software when available and justified by the complexity of the system. The correctness of such tools should be demonstrated by specific testing, by an extensive history of satisfactory use, or by independent verification of their output for the particular safety-related system that is being designed.

Support tools should be selected for their degree of integration. In this context, tools are integrated if they work co-operatively such that the outputs from one tool have suitable content and format for automatic input to a subsequent tool, thus minimizing the possibility of introducing human error in the reworking of intermediate results.

References:

Overview of Technology Computer-Aided Design Tools and Applications in Technology Development, Manufacturing and Design. W. Fichtner, Journal of Computational and Theoretical Nanoscience, Volume 5, Number 6, June 2008, pp. 1089-1105

1620 *The Electromagnetic Data Exchange: Much more than a Common Data Format*. P.E. Frandsen
1621 et al. In *Proceeding of the 2nd European Conference on Antennas and Propagation*. The
1622 Institution of Engineering and Technology (IET), 2007, ISBN 978-0-86341-842-6

1623 *Software engineering: Update*. Ian Sommerville, Addison-Wesley Longman, Amsterdam; 8th
1624 ed., 2006, ISBN 0321313798, 9780321313799

1625 *Software Engineering, Global Edition. 10th edition*, I. Sommerville, Pearson, 2015, ISBN 978-
1626 1292-09613-1.

1627 **B.3.6 Simulation**

1628 NOTE This technique/measure is referenced in Tables B.2, B.5 and B.6 of IEC 61508-2.

1629 **Aim:** To carry out a systematic and complete inspection of an electrical/electronic circuit, of
1630 both the functional performance and the correct dimensioning of the components.

1631 **Description:** The function of the safety-related system circuit is simulated on a computer via a
1632 software behavioural model. Individual components of the circuit each have their own simulated
1633 behaviour, and the response of the circuit in which they are connected is examined by looking
1634 at the marginal data of each component.

1635 **B.3.7 Inspection (reviews and analysis)**

1636 NOTE This technique/measure is referenced in Tables B.2 and B.6 of IEC 61508-2.

1637 **Aim:** To reveal discrepancies between the specification and implementation.

1638 **Description:** Specified functions of the safety-related system are examined and evaluated to
1639 ensure that the safety-related system conforms to the requirements given in the specification.
1640 Any points of doubt concerning the implementation and use of the product are documented so
1641 they may be resolved. In contrast to a walk-through, the author is passive and the inspector is
1642 active during the inspection procedure.

1643 **References:**

1644 IEC 61160:2005, *Design Review*

1645 *Software engineering: Update*. Ian Sommerville, Addison-Wesley Longman, Amsterdam; 8th
1646 ed., 2006, ISBN 0321313798, 9780321313799

1647 *Software Engineering, Global Edition. 10th edition*, I. Sommerville, Pearson, 2015, ISBN 978-
1648 1292-09613-1.

1649 *The Art of Software Testing, third edition*. G. J. Myers, C. Sandler, T. Badgett, John Wiley and
1650 Sons, 2011, ISBN 978-1-118-13315-6

1651 *Software Testing: A Craftsman's Approach, fourth edition*, P. C. Jorgensen, Auerbach
1652 Publications, 2013, ISBN 978-1-4665-6068-0

1653 ANSI/IEEE 1028:1997, *IEEE Standard for software reviews*

1654 *Dependability of Critical Computer Systems 3*. P. G. Bishop et al., Elsevier Applied Science,
1655 1990, ISBN 1-85166-544-7

1656 **B.3.8 Walk-through**

1657 NOTE This technique/measure is referenced in Tables B.2 and B.6 of IEC 61508-2.

Aim: To reveal discrepancies between the specification and implementation.

Description: Specified functions of the safety-related system draft are examined and evaluated to ensure that the safety-related system complies with the requirements given in the specification. Doubts and potential weak points concerning the realisation and use of the product are documented so that they may be resolved. In contrast to an inspection, the author is active and the inspector is passive during the walk-through.

References:

Software engineering: Update. Ian Sommerville, Addison-Wesley Longman, Amsterdam; 8th ed., 2006, ISBN 0321313798, 9780321313799

Software Engineering, Global Edition. 10th edition, I. Sommerville, Pearson, 2015, ISBN 978-1292-09613-1.

ANSI/IEEE 1028:1997, *IEEE Standard for software reviews*

Dependability of Critical Computer Systems 3. P. G. Bishop et al., Elsevier Applied Science, 1990, ISBN 1-85166-544-7

Methodisches Testen von Programmen. G. J. Myers, Oldenbourg Verlag, München, Wien, 1987

B.4 E/E/PE system operation and maintenance procedures

Global objective: To develop procedures which help to avoid failures during the operation and maintenance of the safety-related system.

B.4.1 Operation and maintenance instructions

NOTE This technique/measure is referenced in Table B.4 of IEC 61508-2.

Aim: To avoid mistakes during operation and maintenance of the safety-related system.

Description: User instructions contain essential information on how to use and how to maintain the safety-related system. In special cases, these instructions will also include examples on how to install the safety-related system in general. All instructions should be easily understood. Figures and schematics should be used to describe complex procedures and dependencies.

Reference: *Guidelines for Safe Automation of Chemical Processes.* CCPS, AIChE, New York, 1993, ISBN-10: 0-8169-0554-1, ISBN-13: 978-0-8169-0554-6

B.4.2 User friendliness

NOTE This technique/measure is referenced in Table B.4 of IEC 61508-2 and Table A.12 of IEC61508.

Aim: To reduce complexity during operation of the safety-related system.

Description: The correct operation of the safety-related system may depend to some degree on human operation. By considering the relevant system design and the design of the workplace, the safety-related system developer should ensure that

- the need for human intervention is restricted to an absolute minimum;
- the necessary intervention is as simple as possible;
- the potential for harm from operator error is minimised;
- the intervention facilities and indication facilities are designed according to ergonomic requirements;