

BS EN 61078:2016



BSI Standards Publication

# Reliability block diagrams

**National foreword**

This British Standard is the UK implementation of EN 61078:2016. It is identical to IEC 61078:2016. It supersedes BS EN 61078:2006 which will be withdrawn on 16 September 2019.

The UK participation in its preparation was entrusted to Technical Committee DS/1, Dependability.

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

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

© The British Standards Institution 2016.

Published by BSI Standards Limited 2016

ISBN 978 0 580 87533 5

ICS 03.120.01; 03.120.99

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

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 December 2016.

**Amendments/corrigenda issued since publication**

Date	Text affected
------	---------------

---

EUROPEAN STANDARD

**EN 61078**

NORME EUROPÉENNE

EUROPÄISCHE NORM

November 2016

ICS 03.120.01; 03.120.99

Supersedes EN 61078:2006

English Version

**Reliability block diagrams  
(IEC 61078:2016)**Diagrammes de fiabilité  
(IEC 61078:2016)Zuverlässigkeitsblockdiagramme  
(IEC 61078:2016)

This European Standard was approved by CENELEC on 2016-09-16. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

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

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

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



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

**CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels**

## European foreword

The text of document 56/1685/FDIS, future edition 3 of IEC 61078, prepared by IEC/TC 56 "Dependability" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 61078:2016.

The following dates are fixed:

- latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement (dop) 2017-06-16
- latest date by which the national standards conflicting with the document have to be withdrawn (dow) 2019-09-16

This document supersedes EN 61078:2006.

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

## Endorsement notice

The text of the International Standard IEC 61078:2016 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following notes have to be added for the standards indicated:

IEC 61025	NOTE	Harmonized as EN 61025.
IEC 61165	NOTE	Harmonized as EN 61165.
IEC 62551	NOTE	Harmonized as EN 62551.
IEC 60812	NOTE	Harmonized as EN 60812.
IEC 61508:2010 Series	NOTE	Harmonized as EN 61508:2010 Series.
IEC 61511:2016 Series	NOTE	Harmonized as EN 61511:2016 Series.
ISO/TR 12489	NOTE	Harmonized as CEN ISO/TR 12489.

## Annex ZA (normative)

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

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

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

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

<u>Publication</u>	<u>Year</u>	<u>Title</u>	<u>EN/HD</u>	<u>Year</u>
IEC 60050-192	-	International Electrotechnical Vocabulary - - Part 192: Dependability		-
IEC 61703	-	Mathematical expressions for reliability, availability, maintainability and maintenance support terms	EN 61703	-

## CONTENTS

FOREWORD.....	8
INTRODUCTION.....	10
1 Scope.....	11
2 Normative references.....	11
3 Terms and definitions .....	11
4 Symbols and abbreviated terms .....	18
5 Preliminary considerations, main assumptions, and limitations.....	22
5.1 General considerations.....	22
5.2 Pre-requisite/main assumptions.....	23
5.3 Limitations .....	23
6 Establishment of system success/failed states .....	24
6.1 General considerations.....	24
6.2 Detailed considerations .....	24
6.2.1 System operation .....	24
6.2.2 Environmental conditions .....	25
6.2.3 Duty cycles .....	25
7 Elementary models .....	25
7.1 Developing the model.....	25
7.2 Series structures.....	25
7.3 Parallel structures .....	26
7.4 Mix of series and parallel structures.....	26
7.5 Other structures .....	27
7.5.1 <i>m</i> out of <i>n</i> structures.....	27
7.5.2 Structures with common blocks .....	28
7.5.3 Composite blocks.....	29
7.6 Large RBDs and use of transfer gates .....	29
8 Qualitative analysis: minimal tie sets and minimal cut sets.....	30
8.1 Electrical analogy.....	30
8.2 Series-parallel representation with minimal success path and cut sets.....	32
8.3 Qualitative analysis from minimal cut sets.....	33
9 Quantitative analysis: blocks with constant probability of failure/success .....	33
9.1 Series structures.....	33
9.2 Parallel structures .....	34
9.3 Mix of series and parallel structures.....	34
9.4 <i>m/n</i> architectures (identical items).....	35
10 Quantitative analysis: blocks with time dependent probabilities of failure/success .....	35
10.1 General.....	35
10.2 Non-repaired blocks .....	36
10.2.1 General .....	36
10.2.2 Simple non-repaired block.....	36
10.2.3 Non-repaired composite blocks.....	36
10.2.4 RBDs with non-repaired blocks.....	37
10.3 Repaired blocks .....	37
10.3.1 Availability calculations .....	37
10.3.2 Average availability calculations .....	40

10.3.3	Reliability calculations.....	42
10.3.4	Frequency calculations.....	43
11	Boolean techniques for quantitative analysis of large models.....	43
11.1	General.....	43
11.2	Method of RBD reduction .....	44
11.3	Use of total probability theorem .....	45
11.4	Use of Boolean truth tables .....	46
11.5	Use of Karnaugh maps .....	47
11.6	Use of the Shannon decomposition and binary decision diagrams .....	49
11.7	Use of Sylvester-Poincaré formula.....	50
11.8	Examples of RBD application.....	51
11.8.1	Models with repeated blocks .....	51
11.8.2	$m$ out of $n$ models (non-identical items).....	54
12	Extension of reliability block diagram techniques .....	54
12.1	Non-coherent reliability block diagrams.....	54
12.2	Dynamic reliability block diagrams .....	57
12.2.1	General .....	57
12.2.2	Local interactions.....	58
12.2.3	Systemic dynamic interactions.....	59
12.2.4	Graphical representations of dynamic interactions .....	59
12.2.5	Probabilistic calculations .....	62
Annex A (informative)	Summary of formulae .....	63
Annex B (informative)	Boolean algebra methods.....	67
B.1	Introductory remarks .....	67
B.2	Notation.....	67
B.3	Tie sets (success paths) and cut sets (failure paths) analysis.....	68
B.3.1	Notion of cut and tie sets.....	68
B.3.2	Series-parallel representation using minimal tie and cut sets.....	69
B.3.3	Identification of minimal cuts and tie sets.....	70
B.4	Principles of calculations .....	71
B.4.1	Series structures .....	71
B.4.2	Parallel structures .....	71
B.4.3	Mix of series and parallel structures .....	73
B.4.4	$m$ out of $n$ architectures (identical items).....	73
B.5	Use of Sylvester Poincaré formula for large RBDs and repeated blocks.....	74
B.5.1	General .....	74
B.5.2	Sylvester Poincaré formula with tie sets.....	74
B.5.3	Sylvester Poincaré formula with cut sets.....	76
B.6	Method for disjointing Boolean expressions .....	77
B.6.1	General and background .....	77
B.6.2	Disjointing principle.....	78
B.6.3	Disjointing procedure .....	79
B.6.4	Example of application of disjointing procedure.....	79
B.6.5	Comments .....	81
B.7	Binary decision diagrams .....	82
B.7.1	Establishing a BDD .....	82
B.7.2	Minimal success paths and cut sets with BDDs .....	84
B.7.3	Probabilistic calculations with BDDs .....	86

B.7.4	Key remarks about the use of BDDs .....	87
Annex C (informative)	Time dependent probabilities and RBD driven Markov processes .....	88
C.1	General.....	88
C.2	Principle for calculation of time dependent availabilities .....	88
C.3	Non-repaired blocks .....	89
C.3.1	General .....	89
C.3.2	Simple non-repaired blocks .....	89
C.3.3	Composite block: example on a non-repaired standby system .....	89
C.4	RBD driven Markov processes .....	91
C.5	Average and asymptotic (steady state) availability calculations .....	92
C.6	Frequency calculations.....	93
C.7	Reliability calculations .....	94
Annex D (informative)	Importance factors .....	96
D.1	General.....	96
D.2	Vesely-Fussell importance factor .....	96
D.3	Birnbaum importance factor or marginal importance factor .....	96
D.4	Lambert importance factor or critical importance factor .....	97
D.5	Diagnostic importance factor .....	97
D.6	Risk achievement worth .....	98
D.7	Risk reduction worth.....	98
D.8	Differential importance measure .....	98
D.9	Remarks about importance factors.....	99
Annex E (informative)	RBD driven Petri nets .....	100
E.1	General.....	100
E.2	Example of sub-PN to be used within RBD driven PN models .....	100
E.3	Evaluation of the DRBD state .....	102
E.4	Availability, reliability, frequency and MTTF calculations .....	104
Annex F (informative)	Numerical examples and curves .....	105
F.1	General.....	105
F.2	Typical series RBD structure .....	105
F.2.1	Non-repaired blocks .....	105
F.2.2	Repaired blocks .....	106
F.3	Typical parallel RBD structure .....	107
F.3.1	Non-repaired blocks .....	107
F.3.2	Repaired blocks .....	108
F.4	Complex RBD structures .....	109
F.4.1	Non series-parallel RBD structure.....	109
F.4.2	Convergence to asymptotic values versus MTTR .....	110
F.4.3	System with periodically tested components .....	111
F.5	Dynamic RBD example.....	113
F.5.1	Comparison between analytical and Monte Carlo simulation results .....	113
F.5.2	Dynamic RBD example.....	113
Bibliography	.....	116
Figure 1	– Shannon decomposition of a simple Boolean expression and resulting BDD .....	18
Figure 2	– Series reliability block diagram .....	25
Figure 3	– Parallel reliability block diagram .....	26



Figure 4 – Parallel structure made of duplicated series sub-RBD .....	26
Figure 5 – Series structure made of parallel reliability block diagram.....	27
Figure 6 – General series-parallel reliability block diagram .....	27
Figure 7 – Another type of general series-parallel reliability block diagram .....	27
Figure 8 – 2 out of 3 redundancy.....	28
Figure 9 – 3 out of 4 redundancy.....	28
Figure 10 – Diagram not easily represented by series/parallel arrangement of blocks .....	28
Figure 11 – Example of RBD implementing dependent blocks .....	29
Figure 12 – Example of a composite block.....	29
Figure 13 – Use of transfer gates and sub-RBDs .....	30
Figure 14 – Analogy between a block and an electrical switch.....	30
Figure 15 – Analogy with an electrical circuit .....	31
Figure 16 – Example of minimal success path (tie set).....	31
Figure 17 – Example of minimal failure path (cut set).....	31
Figure 18 – Equivalent RBDs with minimal success paths .....	32
Figure 19 – Equivalent RBDs with minimal cut sets.....	33
Figure 20 – Link between a basic series structure and probability calculations .....	33
Figure 21 – Link between a parallel structure and probability calculations .....	34
Figure 22 – "Availability" Markov graph for a simple repaired block .....	38
Figure 23 – Standby redundancy.....	38
Figure 24 – Typical availability of a periodically tested block.....	39
Figure 25 – Example of RBD reaching a steady state.....	41
Figure 26 – Example of RBD with recurring phases .....	41
Figure 27 – RBD and equivalent Markov graph for reliability calculations .....	42
Figure 28 – Illustrating grouping of blocks before reduction.....	44
Figure 29 – Reduced reliability block diagrams .....	44
Figure 30 – Representation of Figure 10 when item A has failed .....	45
Figure 31 – Representation of Figure 10 when item A is working.....	45
Figure 32 – RBD representing three redundant items.....	46
Figure 33 – Shannon decomposition equivalent to Table 5.....	49
Figure 34 – Binary decision diagram equivalent to Table 5.....	49
Figure 35 – RBD using an arrow to help define system success .....	51
Figure 36 – Alternative representation of Figure 35 using repeated blocks and success paths.....	51
Figure 37 – Other alternative representation of Figure 35 using repeated blocks and minimal cut sets.....	52
Figure 38 – Shannon decomposition related to Figure 35.....	53
Figure 39 – 2-out-of-5 non-identical items .....	54
Figure 40 – Direct and inverted block .....	55
Figure 41 – Example of electrical circuit with a commutator A .....	55
Figure 42 – Electrical circuit: failure paths .....	55
Figure 43 – Example RBD with blocks with inverted states.....	56
Figure 44 – BDD equivalent to Figure 43 .....	57
Figure 45 – Symbol for external elements.....	58

Figure 46 – Dynamic interaction between a CCF and RBDs' blocks.....	60
Figure 47 – Various ways to indicate dynamic interaction between blocks .....	60
Figure 48 – Dynamic interaction between a single repair team and RBDs' blocks .....	60
Figure 49 – Implementation of a PAND gate .....	61
Figure 50 – Equivalent finite-state automaton and example of chronogram for a PAND gate .....	61
Figure 51 – Implementation of a SEQ gate .....	61
Figure 52 – Equivalent finite-state automaton and example of chronogram for a SEQ gate .....	62
Figure B.1 – Examples of minimal tie sets (success paths) .....	68
Figure B.2 – Examples of non-minimal tie sets (non minimal success paths) .....	68
Figure B.3 – Examples of minimal cut sets .....	69
Figure B.4 – Examples of non-minimal cut sets.....	69
Figure B.5 – Example of RBD with tie and cut sets of various order .....	70
Figure B.6 – Reminder of the RBD in Figure 35 .....	82
Figure B.7 – Shannon decomposition of the Boolean function represented by Figure B.6.....	82
Figure B.8 – Identification of the parts which do not matter .....	83
Figure B.9 – Simplification of the Shannon decomposition .....	83
Figure B.10 – Binary decision diagram related to the RBD in Figure B.6.....	84
Figure B.11 – Obtaining success paths (tie sets) from an RBD.....	84
Figure B.12 – Obtaining failure paths (cut sets) from an RBD.....	85
Figure B.13 – Finding cut and tie sets from BDDs .....	85
Figure B.14 – Probabilistic calculations from a BDD.....	86
Figure B.15 – Calculation of conditional probabilities using BDDs .....	87
Figure C.1 – Principle of time dependent availability calculations .....	88
Figure C.2 – Principle of RBD driven Markov processes .....	91
Figure C.3 – Typical availability of RBD with quickly repaired failures .....	91
Figure C.4 – Example of simple multi-phase Markov process .....	92
Figure C.5 – Typical availability of RBD with periodically tested failures .....	92
Figure E.1 – Example of a sub-PN modelling a DRBD block.....	100
Figure E.2 – Example of a sub-PN modelling a common cause failure.....	101
Figure E.3 – Example of DRBD based on RBD driven PN .....	101
Figure E.4 – Logical calculation of classical RBD structures.....	102
Figure E.5 – Example of logical calculation for an $n/m$ gate .....	102
Figure E.6 – Example of sub-PN modelling a PAND gate with 2 inputs .....	103
Figure E.7 – Example of the inhibition of the failure of a block .....	104
Figure E.8 – Sub-PN for availability, reliability and frequency calculations.....	104
Figure F.1 – Availability/reliability of a typical non-repaired series structure .....	105
Figure F.2 – Failure rate and failure frequency related to Figure F.1 .....	106
Figure F.3 – Equivalence of a non-repaired series structure to a single block.....	106
Figure F.4 – Availability/reliability of a typical repaired series structure .....	106
Figure F.5 – Failure rate and failure frequency related to Figure F.4 .....	107
Figure F.6 – Availability/reliability of a typical non-repaired parallel structure .....	107
Figure F.7 – Failure rate and failure frequency related to Figure F.6 .....	108
Figure F.8 – Availability/reliability of a typical repaired parallel structure .....	108

Figure F.9 – Vesely failure rate and failure frequency related to Figure F.8 .....	109
Figure F.10 – Example 1 from 7.5.2 .....	109
Figure F.11 – Failure rate and failure frequency related to Figure F.10.....	110
Figure F.12 – Impact of the MTTR on the convergence quickness .....	111
Figure F.13 – System with periodically tested blocks .....	112
Figure F.14 – Failure rate and failure frequency related to Figure F.13.....	112
Figure F.15 – Analytical versus Monte Carlo simulation results .....	113
Figure F.16 – Impact of CCF and limited number of repair teams .....	114
Figure F.17 – Markov graphs modelling the impact of the number of repair teams .....	115
Figure F.18 – Approximation for two redundant blocks .....	115
Table 1 – Acronyms used in IEC 61078 .....	18
Table 2 – Symbols used in IEC 61078 .....	19
Table 3 – Graphical representation of RBDs: Boolean structures .....	21
Table 4 – Graphical representation of RBDs: non-Boolean structures/DRBD .....	22
Table 5 – Application of truth table to the example of Figure 32 .....	46
Table 6 – Karnaugh map related to Figure 10 when A is in up state .....	48
Table 7 – Karnaugh map related to Figure 10 when A is in down state .....	48
Table 8 – Karnaugh map related to Figure 35 .....	53
Table A.1 – Example of equations for calculating the probability of success of basic configurations .....	63
Table F.1 – Impact of functional dependencies .....	114

# INTERNATIONAL ELECTROTECHNICAL COMMISSION

---

## RELIABILITY BLOCK DIAGRAMS

### 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) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 61078 has been prepared by IEC technical committee 56: Dependability.

This third edition cancels and replaces the second edition published in 2006. This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

- a) the structure of the document has been entirely reconsidered, the title modified and the content extended and improved to provide more information about availability, reliability and failure frequency calculations;
- b) Clause 3 has been extended and clauses have been introduced to describe the electrical analogy, the "non-coherent" RBDs and the "dynamic" RBDs;
- c) Annex B about Boolean algebra methods has been extended;
- d) Annex C (Calculations of time dependent probabilities), Annex D (Importance factors), Annex E (RBD driven Petri net models) and Annex F (Numerical examples and curves) have been introduced.

The text of this standard is based on the following documents:

FDIS	Report on voting
56/1685/FDIS	56/1694/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

## INTRODUCTION

A reliability block diagram (RBD) is a pictorial representation of a system's successful functioning. It shows the logical connection of (functioning) components (represented by blocks) needed for successful operation of the system (hereafter referred to as "system success"). Therefore an RBD is equivalent to a logical equation of Boolean variables and the probabilistic calculations are primarily related to constant values of the block success/failure probabilities.

Many different analytical methods of dependability analysis are available, of which the RBD is one. Therefore, the purpose of each method and their individual or combined applicability in evaluating the availability, reliability, failure frequency and other dependability measures as may be applicable to a given system or component should be examined by the analyst prior to deciding to use the RBD. Consideration should also be given to the results obtainable from each method, data required to perform the analysis, complexity of analysis and other factors identified in this standard.

Provided that the blocks in the RBD behave independently from each other and that the order in which failures occur does not matter then the probabilistic calculations can be extended to time dependent probabilistic calculations involving non-repaired as well as repaired blocks (e.g. blocks representing non-repaired or repaired components). In this case three dependability measures related to the system successful functioning have to be considered: the reliability itself,  $R_S(t)$ , but also the availability,  $A_S(t)$  and the failure frequency,  $w_S(t)$ . While, for systems involving repaired components, the calculations of  $A_S(t)$  or  $w_S(t)$  can be done quite straightforwardly, the calculation of  $R_S(t)$  implies systemic dependencies (see definition 3.34) which cannot be taken into account within the mathematical framework of RBDs. Nevertheless, in particular cases, approximations of  $R_S(t)$  are available.

The RBD technique is linked to fault tree analysis [1]<sup>1</sup> and to Markov techniques [2]:

- The underlying mathematics is the same for RBDs and fault tree analysis (FTA): when an RBD is focused on system success, the FT is focused on system failure. It is always possible to transform an RBD into an FT and vice versa. From a mathematical point of view, RBD and FT models share dual logical expressions. Therefore, the mathematical developments and the limitations are similar in both cases.
- When the availability  $A_i(t)$  of one block can be calculated by using an individual Markov process [2] independent of the other blocks, this availability,  $A_i(t)$ , can be used as input for the calculations related to an RBD including this block. This approach where an RBD provides the logic structure and Markov processes numerical values of the availabilities of the blocks is called "RBD driven Markov processes".

For systems where the order of failures is to be taken into account, or where the repaired blocks do not behave independently from each other or where the system reliability,  $R_S(t)$ , cannot be calculated by analytical methods, Monte Carlo simulation or other modelling techniques, such as dynamic RBDs, Markov [2] or Petri net techniques [3], may be more suitable.

---

<sup>1</sup> Numbers in square brackets refer to the Bibliography.

# RELIABILITY BLOCK DIAGRAMS

## 1 Scope

This International Standard describes:

- the requirements to apply when reliability block diagrams (RBDs) are used in dependability analysis;
- the procedures for modelling the dependability of a system with reliability block diagrams;
- how to use RBDs for qualitative and quantitative analysis;
- the procedures for using the RBD model to calculate availability, failure frequency and reliability measures for different types of systems with constant (or time dependent) probabilities of blocks success/failure, and for non-repaired blocks or repaired blocks;
- some theoretical aspects and limitations in performing calculations for availability, failure frequency and reliability measures;
- the relationships with fault tree analysis (see IEC 61025 [1]) and Markov techniques (see IEC 61165 [2]).

## 2 Normative references

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

IEC 60050-192, *International Electrotechnical Vocabulary – Part 192: Dependability* (available at <http://www.electropedia.org>)

IEC 61703, *Mathematical expressions for reliability, availability, maintainability and maintenance support terms*

## 3 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 60050-192 as well as the following apply.

NOTE Some terms have been taken from IEC 60050-192 and modified for the needs of this standard.

### 3.1

#### **reliability block diagram**

#### **RBD**

logical, graphical representation of a system showing how the success states of its sub-items (represented by blocks) and combinations thereof, affect system success state

Note 1 to entry: The RBD technique was developed a long time ago when the term “reliability” was used as an umbrella term for “successful functioning”. This umbrella term is now superseded by “dependability”. Nevertheless it is still in use in the vernacular language and terms like “reliability engineering”, “reliability studies” or “reliability block diagram”. Therefore the term “reliability” used in RBD does not mean that this technique allows to calculate the reliability of a complex system straightforwardly from reliabilities of its constituting blocks (see 10.3.1.4).

Note 2 to entry: An RBD is a directed acyclic graph (i.e. a graph without loops) representing the logical links between the success state of a system and the success states of its constituting blocks. This logical architecture is mainly represented by conventional series and parallel graphical structures (see Clause 4 and Clause 7).

Note 3 to entry: RBDs may be extended to represent multi-state (i.e. more than two states) systems but those extensions cannot be handled within the Boolean framework.

[SOURCE: IEC 60050-192:2015, 192-11-03, modified – Notes added]

### 3.2

#### **Boolean related model**

mathematical model where the state of a system is represented by a logical function of Boolean variables representing the states of its components

Note 1 to entry: A Boolean variable  $a$  only has two values and a logical function of several Boolean variables also has only two values. Those two values may be for example, {0, 1}, {up, down}, {true, false}, {working, failed}, etc. The underlying mathematics behind the logical functions is Boolean algebra.

### 3.3

#### **RBD driven Markov process**

Markov process modelled by an RBD made of blocks modelled by individual sub-Markov models behaving independently from each other

Note 1 to entry: The underlying logic of an RBD allows to combine the individual availabilities of the blocks to obtain the system availability. When the blocks are modelled by small individual Markov processes (e.g. with less than 10 states) the RBD is equivalent to the Markov process related to the system which may encompass millions of states. This is the basis for most of the probabilistic calculations achieved with RBDs. Such Markov process built through the use of the RBD as guideline is called "RBD driven Markov process".

Note 2 to entry: The independent Markov process is developed in [2].

### 3.4

#### **dynamic RBD**

##### **DRBD**

reliability block diagram where the assumption of independency between the blocks is not fulfilled

Note 1 to entry: The blocks of a DRBD can have interactions with elements external to the RBD itself.

### 3.5

#### **non-coherent RBD**

reliability block diagram modelling a non-monotonic logical function

Note 1 to entry: A non-coherent RBD is an RBD where the blocks may appear both in direct and inverted states (see Table 3). In this case, some of the minimal success path (see definition 3.15) may have some blocks in down state and some minimal failure paths, some blocks in up state. The concepts of minimal tie and cut sets are no longer valid and have to be replaced by the concept of prime implicants.

Note 2 to entry: In a non-coherent RBD, a minimal success path may become a failure path by the repair of a block in down state and a minimal failure path may become a success path by a further failure of one block in up state. This is why they are named "non-coherent".

### 3.6

#### **item**

subject being considered

Note 1 to entry: In this International Standard the word "item" covers mainly the system modelled by the RBD and the "blocks" in the RBD.

[SOURCE: IEC 60050-192:2015, 192-01-01, modified – Notes to entry have been deleted, Note 1 to entry has been added]

### 3.7

#### **block**

basic element used to build an RBD

Note 1 to entry: A block has only two states (up and down) and may represent any item with two states (e.g. components, functions, subsystems) repaired or not repaired. By analogy and to simplify the wording, a repaired/non-repaired block represents a repaired/non-repaired item, the failure/repair of a block represents the



failure/repair of the modelled item and the up/down state of a block represents the up/down state of the modelled item.

Note 2 to entry: The number of states may be extended to more than two states in order to represent multi-state (i.e. more than two states) systems but those extensions cannot be handled within the Boolean framework.

Note 3 to entry: For the purposes of this standard, the blocks are divided between "elementary blocks" – or more simply, "blocks" – and "composite blocks" comprising several "elementary blocks". This is illustrated in Table 3.

### 3.8

#### **repeated block**

block appearing more than once in an RBD

Note 1 to entry: Repeated blocks represent the same physical items. This should not be confused with duplicated blocks which represent different but similar physical items used to implement redundancy.

Note 2 to entry: Repeated blocks can appear in the direct or inverted state (i.e.; the block appears in up state in a part of the RBD and down state in another part, or vice versa). They are very useful to represent RBDs related to a complex system or for representing RBDs in the form of success or failure paths (see 8.2).

### 3.9

#### **up state**

#### **available state**

state of being able to perform as required

Note 1 to entry: The absence of necessary external resources may prevent operation, but do not affect the up state.

Note 2 to entry: Up state relates to availability of the item.

Note 3 to entry: An item may be considered to be in an up state for some functions and in a down state for others, concurrently.

Note 4 to entry: The adjectives "up" and "available" designate an item in an up state.

Note 5 to entry: Within the context of RBDs, the state of a block is identical to the state of the component modelled by this block. Therefore a block in up state refers to a component in up state. The same concept applies to the RBD and the corresponding system.

Note 6 to entry: Within an RBD and by analogy with an electrical circuit, a block in the up state can be considered as a virtual switch in closed position and a block in the down state as a virtual switch in open position.

[SOURCE: IEC 60050-192:2015, 192-02-01, modified – Note 5 to entry and Note 6 to entry have been added]

### 3.10

#### **up time**

time interval for which the item is in an up state

[SOURCE: IEC 60050-192:2015, 192-02-02]

### 3.11

#### **mean up time**

#### **MUT**

expectation of the up time

[SOURCE: IEC 60050-192:2015, 192-08-09]

### 3.12

#### **down state**

#### **unavailable state**

state of being unable to perform as required, due to internal fault, or preventive maintenance

Note 1 to entry: "Down" state relates to unavailability of the item.

Note 2 to entry: The adjectives "down" or "unavailable" designate an item in a down state.

Note 3 to entry: Within the context of RBDs, the state of a block (respectively an RBD) is assimilated to the state of the component (respectively the system) modelled by this block (respectively this RBD). Therefore a block (respectively an RBD) in down state refers to a component (respectively a system) in down state.

Note 4 to entry: Within an RBD, a block in the down state may be interpreted as an open electrical switch.

[SOURCE: IEC 60050-192:2015, 192-02-20, modified – Note 3 and 4 to entry have been added]

### **3.13 down time**

time interval for which the item is in a down state

[SOURCE: IEC 60050-192:2015, 192-02-21, modified – the figure and Note 1 to entry have been deleted]

### **3.14 mean down time MDT**

expectation of the down time

[SOURCE: IEC 60050-192:2015, 192-08-10]

### **3.15 success path tie set**

set of blocks, with each block in the set being in the up state thus resulting in the RBD to be in the up state

Note 1 to entry: The name tie set is given by analogy with an electrical circuit: the blocks in up states constitute a closed (tied) circuit between the RBD input and the RBD output.

### **3.16 minimal tie set**

tie set such as that any failure of one of the blocks in up state also fails the whole RBD

Note 1 to entry: In a minimal tie set, every block in up state is necessary to retain the RBD in up state.

Note 2 to entry: The order of a minimal tie set is given by the number of blocks in the set in up state: order 1 comprises 1 block in up state, order 2 comprises 2 blocks in up state, etc.

### **3.17 failure path cut set**

set of blocks, with each block in the set being in the down state thus resulting in the RBD to be in the down state

Note 1 to entry: The name cut set is given by analogy with an electrical circuit: the blocks in down state constitute an open (cut) circuit between the RBD input and the RBD output.

### **3.18 minimal cut set**

cut set such as that any restoration of one of the blocks in down state also restore the RBD to up state

Note 1 to entry: In a minimal cut set, every block in down state is necessary to retain the RBD in down state

Note 2 to entry: The order of a minimal cut set is given by the number of blocks in the down state: order 1 comprises 1 block in down state, order 2 comprises 2 blocks in down state, etc.

### **3.19 disjoint set of elements**

set of Boolean elements whose intersections are empty

EXAMPLE If  $(C_i^d)$  is a set of disjoint cut sets then  $C_i^d \cap C_j^d = \Phi \quad \forall i \neq j$  and therefore, the probability  $P(C_i^d \cap C_j^d) = 0 \quad \forall i \neq j$ .

Note 1 to entry: The term "element" is used here in the general meaning used in the set theory, i.e. a member of a given collection of objects.

Note 2 to entry: Disjoint elements are incompatible: when one is true, the other is false and vice versa. This describes mutual exclusiveness and, therefore, complete dependency between the elements.

### 3.20

#### availability

<item> ability to be in up state

[SOURCE: IEC 60050-192:2015, 192-01-23, modified – notes have been deleted]

### 3.21

$A(t)$

#### instantaneous availability

#### point availability

<measure> probability of being in up state at a given instant

[SOURCE: IEC 60050-192:2015, 192-08-01, modified]

### 3.22

$U(t)$

#### instantaneous unavailability

#### point unavailability

<measure> probability of being in down state at a given instant

[SOURCE: IEC 60050-192:2015, 192-08-04, modified]

### 3.23

$A^{avg}(t_1, t_2)$

#### mean availability

#### average availability

<measure> average value of the instantaneous availability over a given time interval  $[t_1, t_2]$

[SOURCE: IEC 60050-192:2015, 192-08-05, modified – Note 1 to entry has been deleted]

### 3.24

$U^{avg}(t_1, t_2)$

#### mean unavailability

#### average unavailability

<measure> average value of the instantaneous unavailability over a given time interval  $(t_1, t_2)$

Note 1 to entry: The mean unavailability of a safety instrumented system (see IEC 61508 [5]) is also called "average probability of failure on demand" (Acronym: PFD<sub>avg</sub>).

[SOURCE: IEC 60050-192:2015, 192-08-06, modified – Note 1 to entry has been replaced]

### 3.25

$A^{st}$

$A^{as}$

#### steady state availability

#### asymptotic availability

limit, if it exists, of the instantaneous availability, when the time tends to infinity

Note 1 to entry: Under certain conditions, the steady state availability may be expressed as the quotient  $MUT/(MUT+MDT)$ . See IEC 61703.

[SOURCE: IEC 60050-192:2015, 192-08-07, modified – Note 1 to entry modified]

### 3.26 reliability

<item> ability to perform as required, without failure, for a given time interval, under given conditions

[SOURCE: IEC 60050-192:2015, 192-01-24, modified – Notes to entry have been deleted]

### 3.27

$R(t_1, t_2)$   
 $R(t)$

#### reliability

<measure> probability of performing as required for time interval  $[t_1, t_2]$ , under given conditions

Note 1 to entry: The reliability  $R(t)$  is the reliability for the time interval  $[0, t]$ .

[SOURCE: IEC 60050-192:2015, 192-01-24, modified – Notes to entry were replaced by new Note 1 to entry]

### 3.28

$F(t_1, t_2)$   
 $F(t)$

#### unreliability

<measure> probability of not performing as required for time interval  $[t_1, t_2]$ , under given conditions

Note 1 to entry: The unreliability  $F(t)$  is the unreliability for the time interval  $[0, t]$ .

Note 2 to entry: The unreliability is the complement to 1 of the reliability:  $F(t)=1-R(t)$ .

### 3.29

$\lambda(t)$

#### instantaneous failure rate failure rate

limit, if it exists, of the quotient of the conditional probability that an item goes from up state to down state within time interval  $[t, t + \Delta t]$ , and  $\Delta t$ , when  $\Delta t$  tends to zero, given that it has not been in down state within time interval  $[0, t]$

Note 1 to entry: The definition has been adapted from IEC 60050-192 to also cover repairable items:

- if the item has no internal built-in redundancy, the failure rate is identical to what it would be if it was not repairable;
- if the item has built-in internal redundancy it can remain in up state when some redundant parts are failed. Therefore, those failures are repairable as long as the whole item has no transition to the down state due to a further part failure.

Note 2 to entry: The terms failure rate (3.29), conditional failure intensity (3.30) and unconditional failure intensity (3.31) seem similar but they differ by the conditional events used in their definitions. Even if these parameters can have close numerical values in particular cases, they behave in different ways and should not be confused with each other.

[SOURCE: IEC 60050-192:2015, 192-05-06, modified – Notes to entry have been replaced by new notes to entry]

**3.30** $\lambda_v(t)$ **instantaneous conditional failure intensity****conditional failure intensity****Vesely failure rate**

limit, if it exists, of the quotient of the conditional probability that the failure of an item occurs within time interval  $[t, t + \Delta t]$ , and  $\Delta t$ , when  $\Delta t$  tends to zero, given that the item was in up state at time  $t$  and at time 0

Note 1 to entry: See Note 2 to entry of the failure rate definition (3.29).

**3.31** $w(t)$ **instantaneous unconditional failure intensity****unconditional failure intensity****failure frequency**

limit, if it exists, of the quotient of the conditional probability that the failure of an item occurs within time interval  $[t, t + \Delta t]$ , and  $\Delta t$ , when  $\Delta t$  tends to zero, given that the item was in up state at time 0

Note 1 to entry: See Note 2 to entry of the failure rate definition (3.29).

Note 2 to entry: This parameter is equivalent to the failure intensity defined in IEC 60050-192:2015, 192-05-08. The name has been modified to distinguish it from the term, conditional failure intensity (3.30).

**3.32** $w^{\text{avg}}(0, T)$ **average failure frequency**

number of failures per unit of time of an item averaged over a given period of time  $T$

Note 1 to entry: If  $N$  is the number of failures of the item over  $[0, T]$  then the average failure frequency over this period of time is calculated as  $w^{\text{avg}}(0, T) = N/T$ .

Note 2 to entry: If  $m$  is the mean time between failures (see IEC 60050-192) of an item then the average number of failures occurring over  $[0, T]$  is  $N \approx T/m$ . Therefore  $w^{\text{avg}}(0, T) = N/T \approx 1/m$ .

Note 3 to entry: Mathematically speaking,  $w^{\text{avg}}(0, T)$  is the average of  $w(t)$  over  $[0, T]$ . Then

$$w^{\text{avg}}(0, T) = \frac{1}{T} \int_0^T w(\tau) d\tau.$$

**3.33****mean operating time to failure****MTTF**

expectation of operating time to failure

Note 1 to entry: In the case of non-repairable items with an exponential distribution of times to failure (thus a constant failure rate) the MTTF is numerically equal to the reciprocal of the failure rate. This is also true for repairable items if, after restoration, they can be considered to be “as-good-as-new”.

Note 2 to entry: This note only applies to the French language.

[SOURCE IEC 60050-192, 192-05-11, modified – Note 2 to entry has been deleted]

**3.34****systemic dependency****holistic dependency**

dependency between the parts of a system which are related to the system considered as a whole

EXAMPLE 1 A single repair team constitutes a systemic dependency between repairable items: when an item fails it can be repaired only if the repair team is not busy due to the repair of another item belonging to the system.

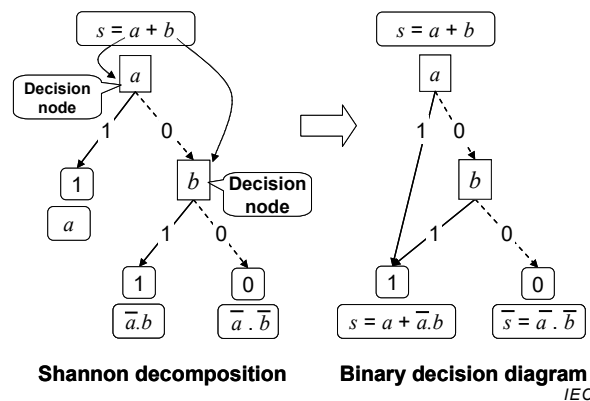
EXAMPLE 2 The reliability  $R(t)$  of a system can be expressed as the probability for the system to be in up state at time  $t$ , provided it has never been in down state over the interval  $[0, t]$ . Therefore only the sequences of events

which do not lead to the down state over  $[0, t]$  can be retained and the sequences of events which include a succession "up state"→"down state"→"up state" have to be excluded from the calculations. This implies that, with regards to the calculation of  $R(t)$ , an item going to the down state is repairable only if the system remains in up state during the repair of the item. Therefore, with regards to the calculation of  $R(t)$ , the items are repairable or not depending on the states of the other blocks and this constitutes systemic dependencies between all blocks of the RBD modelling the system.

Note 1 to entry: A systemic dependency cannot be described as a local property of the individual items of the system.

### 3.35 binary decision diagram BDD

compact decision tree based on the Shannon decomposition of a Boolean expression



**Figure 1 – Shannon decomposition of a simple Boolean expression and resulting BDD**

Note 1 to entry: Figure 1 illustrates how the simple Boolean expression  $s = a + b$  can be transformed into a decision tree by using the Shannon decomposition and then how the corresponding BDD is obtained by gathering the paths giving the same value (0 or 1) of the Boolean expression.

Note 2 to entry: Mathematically speaking, BDDs are rooted, directed acyclic graphs. This is a data structure expressing Boolean expressions as unions of disjointed terms. This, in turn, leads to exact probabilistic calculations. This is the state of the art with regards to probabilistic calculation on Boolean related models. More details about BDDs can be found in reference [33].

Note 3 to entry: This note only applies to the French language.

## 4 Symbols and abbreviated terms

**Table 1 – Acronyms used in IEC 61078**

Abbreviation/Acronym	Meaning
BDD	Binary decision diagram.
CCF	Common cause failure.
FMEA	Failure modes and effects analysis.
FT, FTA	Fault tree, fault tree analysis.
MTTF	Mean operating time to failure.
MTTR	Mean time to restoration.
DRBD	Dynamic reliability block diagram.
$PFD_{avg}$	Average of the probability of failure on demand (mean unavailability).
PAND	Priority AND gate.
PN	Petri net.
RBD	Reliability block diagram.
SEQ	Sequential gate.

**Table 2 – Symbols used in IEC 61078**

Symbol	Meaning
S	System modelled by an RBD.
$X \in (A, B, C, \dots), X \neq S$	Blocks used within an RBD. S is reserved for the system and the other letters for the blocks.
$s = S$ in "up" state	Boolean variable indicating that the system S is in the up state. This is also the event "S in up state".
$\bar{s} = S$ in "down" state	Boolean variable indicating that the system S is in the down state. This is also the event "S in down state".
$x = X$ in "up" state	Boolean variable indicating that block X is in the up state. This is also the event "X in up state".
$\bar{x} = X$ in "down" state	Boolean variable indicating that block X is in the down state. This is also the event "X in down state".
$(\Pi_i), (C_i)$	Minimal success paths (minimal tie sets), minimal failure paths (minimal cut sets).
$(\Pi_i^d), (C_i^d)$	Disjoint success paths (disjoint tie sets), disjoint failure paths (disjoint cut sets).
$P(\cdot)$	Probability function.
$P_s = P(S \text{ in "up" state})$	Constant probability that the system S is in the up state.
$P_{\bar{s}} = P(S \text{ in "down" state})$	Constant probability that the system S is in the down state.
$P_x = P(X \text{ in "up" state})$	Constant probability that block X is in the up state.
$P_{\bar{x}} = P(X \text{ in "down" state})$	Constant probability that block X is in the down state.
$P_{s x} = P(S \text{ in "up" state} \mid X \text{ in "up" state})$	Conditional probability that the system S is in the up state given that block X is in the up state.
$P_{s \bar{x}} = P(S \text{ in "up" state} \mid X \text{ in "down" state})$	Conditional probability that the system S is in the up state given that block X is in down state.
$P_s(t)$	Time dependent probabilities that the system S is in the up state.
$P_x(t)$	Time dependent probability that block X is in the up state.
$P_{\bar{s}}(t)$	Time dependent probabilities that the system S is in the down state.
$P_{\bar{x}}(t)$	Time dependent probability that block X is in the down state.
$P(OK, t)$	Probability of state OK at time $t$ .
$t, t_i$	Current instant of time.
$T, T_i$	Time duration.
$[t_1, t_2], [0, T] \equiv [t_1 = 0, t_2 = 0 + T]$	Time interval, $t_1 < t_2$
$A_S(t) = P(S \text{ in "up" state at time } t)$	Availability of the system S at time $t$ .
$A_S^{\text{avg}}(t_1, t_2), A_S^{\text{avg}}(0, T), A_S^{\text{avg}}(T)$	Average availability of the system S over the time interval $[t_1, t_2]$ or $[0, T]$ .
$A_S^{\text{avg}}, A_S^{\text{st}}, A_S^{\text{as}}$	Average availability of the system S over $[0, \infty]$ , steady state availability and asymptotic availability.
$A_X(t) = P(X \text{ in "up" state at time } t)$	Availability of block X at time $t$ .
$A_i(t) = A_{X_i}(t) = P(X_i \text{ in "up" state at time } t)$	Availability of block $X_i$ at time $t$ .
$A_X^{\text{avg}}(t_1, t_2), A_X^{\text{avg}}(0, T), A_X^{\text{avg}}(T)$	Average availability of block X over the time interval $[t_1, t_2]$ or $[0, T]$ .

Symbol	Meaning
$A_X^{avg}, A_X^{st}, A_X^{as}$	Average availability of block X over $[0, \infty]$ , steady state availability and asymptotic availability.
$U_S(t) = P(\text{S in "down" state at time } t)$	Unavailability of the whole system S at time $t$ .
$U_S^{avg}(t_1, t_2), U_S^{avg}(0, T), U_S^{avg}(T)$	Average unavailability of the system S over the time interval $[t_1, t_2]$ or $[0, T]$ .
$U_S^{avg}, U_S^{st}, U_S^{as}$	Average unavailability of the system S over $[0, \infty]$ , steady state unavailability and asymptotic unavailability.
$U_X(t) = P(\text{X in "down" state at time } t)$	Unavailability of block X at time $t$ .
$U_i(t) = U_{X_i}(t) = P(X_i \text{ in "down" state at time } t)$	Unavailability of block $X_i$ at time $t$ .
$U_X^{avg}(t_1, t_2), U_X^{avg}(0, T), U_X^{avg}(T)$	Average unavailability of block X over the time interval $[t_1, t_2]$ or $[0, T]$ .
$U_X^{avg}, U_X^{st}, U_X^{as}$	Average unavailability of block X over $[0, \infty]$ , steady state unavailability and asymptotic unavailability.
$R_S(t) = P(\text{S in "up" state all over } [0, t])$	Reliability of the system S over $[0, t]$ .
$F_S(t) = 1 - R_S(t)$	Unreliability of the overall system S over $[0, t]$ (failure distribution of the system S).
$f_S(t)$	Time to failure density functions of system S.
$R_X(t) = P(\text{X in "up" state all over } [0, t])$	Reliability of block X over $[0, t]$ .
$F_X(t) = 1 - R_X(t)$	Unreliability of block X over $[0, t]$ (failure distribution of block X).
$f_X(t)$	Time to failure density functions of block X.
$\Lambda_S, \Lambda_S(t)$	Constant and time dependent failure rates of the system S.
$\Lambda_{VS}, \Lambda_{VS}(t)$	Conditional failure intensity (Vesely failure rate) of the overall system.
$w_S(t)$	Unconditional failure intensity (failure frequencies) of the system S, at time $t$ .
$W_S(0, T), W_S(T)$	Expected number of failures of the system S over $[0, T]$ .
$w_S^{avg}(0, T), w_S^{avg}(T)$	Average unconditional failure intensity (average failure frequency) of the system S over $[0, T]$ .
$\lambda_X, \lambda_X(t)$	Constant and time dependent failure rates of block X.
$w_X(t)$	Unconditional failure intensity (failure frequencies) of block X, at time $t$ .
$W_X(0, T), W_X(T)$	Expected number of failures of block X over $[0, T]$ .
$w_X^{avg}(0, T), w_X^{avg}(T)$	Average unconditional failure intensity (average failure frequency) of block X over $[0, T]$ .
$\lambda_{Xd}$	Dormant failure rate of block X.
$\mu_X, \mu_X(t)$	Constant or time-dependent repair rates of block X.
$\binom{n}{r}$	Number of ways of selecting $r$ blocks from $n$ blocks without order: $\binom{n}{r} = \frac{n!}{r!(n-r)!}$
"0", "1"	Symbols used in truth tables, Karnaugh map, Shannon decomposition and binary decision diagrams to denote down (failure) states and up (success) states of blocks or of systems.
$\cap, \bullet$	Boolean operators denoting AND logic, e.g. $a \cap b, a \bullet b$ (intersection).

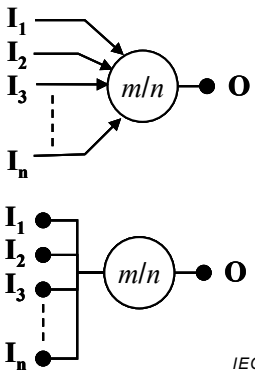


Symbol	Meaning
$\cup, +$	Boolean operators denoting OR logic, e.g. $a \cup b, a + b$ (union).
$\Phi, \Omega$	"Impossible" event and "certain" events.

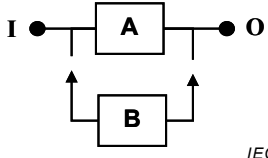
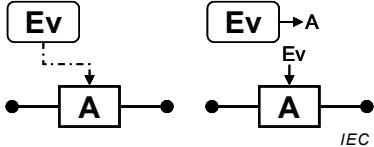
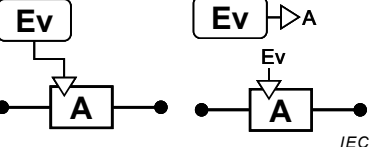
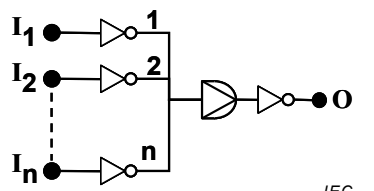
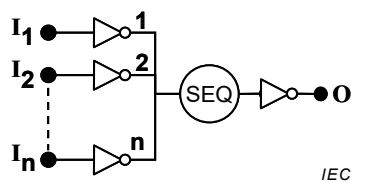
The use of the symbols in Table 3 is recommended when drafting a reliability block diagram (RBD).

**Table 3 – Graphical representation of RBDs: Boolean structures**

Graphical representation	Meaning
<p style="text-align: center;"><b>I</b></p> <p style="text-align: center;"><b>O</b> IEC</p>	<p>Indicates input.</p> <p>Indicates output.</p> <p>Such indications are used for convenience. They are not mandatory, but may be useful where connections have a directional significance.</p>
	<p>An RBD is a directed graph. The direction of each link is from input to output (i.e., from left to right). When needed, arrows may be added to avoid confusion.</p>
	<p>(Elementary) block: grouping of equipment, components, units or other system elements.</p>
	<p>Series structure: the system is up if A and B are in up states.</p> <p>This represents the logic functions <math>s = a \cap b</math>. From a failure point of view it is equivalent to <math>\bar{s} = \bar{a} \cup \bar{b}</math></p>
	<p>Parallel structure (full active redundancy): the system is up if A or B are in up state.</p> <p>This represents the logic functions <math>s = a \cup b</math>. From a failure point of view it is equivalent to <math>\bar{s} = \bar{a} \cap \bar{b}</math></p>
	<p>NOT gate: the output of the gate is equal to 0 when its input is equal to 1 and vice-versa.</p>
	<p>Transfer gates: the output <math>i</math> is linked to the input(s) with the same name. This is useful to:</p> <ul style="list-style-type: none"> <li>– split large RBDs into several smaller parts (sub-RBDs);</li> <li>– transfer the output at one place in an RBD to another place in the RBD.</li> </ul>
	<p>Composite block: grouping of elementary blocks. This may be useful to simplify the RBD drawing, to indicate parts needing further development or to gather non independent individual blocks into a structure independent of the rest of the RBD.</p>
<p style="text-align: center;">Direct state</p> <p style="text-align: center;">Inverted state</p>	<p>Repeated blocks: the same block representing a given item appears in several places of the RBD either in the direct state or the inverted state i.e. when the block A in direct state is "up", the block in the inverted state is "down" and vice versa.</p> <p>These symbols are used for non-coherent RBDs.</p>
	<p>External element interacting with one or several blocks of the RBD.</p> <p>This symbol is used for dynamic RBDs.</p>

Graphical representation	Meaning
 <p>The diagram shows two symbols for majority vote logic. The top symbol is a circle labeled 'm/n' with 'n' inputs (I<sub>1</sub>, I<sub>2</sub>, I<sub>3</sub>, ..., I<sub>n</sub>) and one output 'O'. The bottom symbol is a circle labeled 'm/n' with 'n' inputs (I<sub>1</sub>, I<sub>2</sub>, I<sub>3</sub>, ..., I<sub>n</sub>) and one output 'O'. The IEC logo is at the bottom right of the diagram.</p>	<p>Success majority vote logic (symbol <math>m/n</math>): at least <math>m</math>-out-of-<math>n</math> blocks are needed for system success in an active redundant configuration.</p> <p>NOTE It is important to make the difference between the "success" majority vote logics implemented in RBDs and the "failure" majority vote logics implemented in fault trees. Generally the context (RBD or fault tree) is sufficient to show the difference. The relationship is the following:  <math>(m/n)_{Suc} \equiv ([n-m+1]/n)_{Fail}</math>.</p>

**Table 4 – Graphical representation of RBDs: non-Boolean structures/DRBD**

Graphical representation	Meaning
 <p>The diagram shows a block 'A' with input 'I' and output 'O'. A block 'B' is connected to the output of 'A' and has a feedback loop back to the input of 'A'. The IEC logo is at the bottom right of the diagram.</p>	<p>Standby redundancy: B takes over the function of A when A fails.</p>
 <p>The diagram shows two symbols. The first is a block 'A' with an event 'Ev' pointing to it. The second is a block 'A' with an event 'Ev' pointing to it and an arrow from 'Ev' to the output of 'A'. The IEC logo is at the bottom right of the diagram.</p>	<p>Functional dependencies: the state of A depends on the event Ev. This event may be external or internal to the RBD. This symbol reminds that a dependency exists but the type of dependency may be diverse and has to be described somewhere else.</p>
 <p>The diagram shows two symbols. The first is a block 'A' with an event 'Ev' pointing to it. The second is a block 'A' with an event 'Ev' pointing to it and an arrow from 'Ev' to the output of 'A'. The IEC logo is at the bottom right of the diagram.</p>	<p>Complete functional dependency: when the event Ev occurs, then block A goes to the down state. This event may be external or internal to the RBD. It plays the role of the trigger used in similar structures implemented in dynamic fault trees.</p>
 <p>The diagram shows a PAND gate with 'n' inputs (I<sub>1</sub>, I<sub>2</sub>, ..., I<sub>n</sub>) and one output 'O'. The IEC logo is at the bottom right of the diagram.</p>	<p>PAND gate: the output goes to the down state when the inputs go to the down states in the order <math>I_1</math>, then <math>I_2</math>, then <math>I_3</math>, ... then <math>I_n</math>. The inputs <math>I_1, I_2, I_3, \dots, I_n</math> behave independently from each other.</p> <p>This gate has been introduced to be used in dynamic fault trees and this is why NOT gates are used to invert the inputs and the output in order to be consistent with the RBD logic.</p>
 <p>The diagram shows a SEQ gate with 'n' inputs (I<sub>1</sub>, I<sub>2</sub>, ..., I<sub>n</sub>) and one output 'O'. The IEC logo is at the bottom right of the diagram.</p>	<p>SEQ gate: the output goes to the down state when the inputs go to the down states in the order <math>I_1</math>, then <math>I_2</math>, then <math>I_3</math>, ... then <math>I_n</math>. The inputs don't behave independently as <math>I_n</math> cannot go to the down state if <math>I_{n-1}</math> is not already in the down state, <math>I_{n-1}</math> cannot go to the down state if <math>I_{n-2}</math> is not already in the down state, etc., <math>I_2</math> cannot go to the down if <math>I_1</math> is not already in the down state.</p> <p>This gate has been introduced to be used in dynamic fault trees and this is why NOT gates are used to invert the inputs and the output in order to be consistent with the RBD logic.</p>

## 5 Preliminary considerations, main assumptions, and limitations

### 5.1 General considerations

An RBD models a system using the logical links existing between the success state (up state) of the system (i.e., the overall RBD) and the success states (up states) of its components

(i.e., the blocks of the RBD). Therefore, an RBD embeds a logical formula and this is why an RBD is not necessarily similar to the physical architecture of the system (e.g. two redundant isolation valves in series on the same pipe are represented by two blocks in parallel into the corresponding RBD).

An RBD can be firstly used for qualitative analysis purposes by identifying the combinations of the blocks in the up state allowing the system to be in an up state (success paths or tie sets) or the combinations of the blocks being in down states leading to the system down state (failure paths or cut sets).

Secondly an RBD can be used for probabilistic calculations and, as this is a static representation (i.e., independent of the time), the probabilistic rules are basically related to blocks with constant probabilities of success or failure.

This can be extended to time dependent probabilistic calculations. This may be difficult for reliability calculations but, for availability and frequency calculations and provided that the blocks behave independently from each other, there is no restriction, other than mathematical tractability, on the distribution that may be used to describe the times to failure or repair of the blocks. This allows, for example, to model the (un)availabilities of each of the blocks by individual analytical formulae whose results are combined through the logic of the RBD to obtain the system (un)availability. When those analytical formulae are obtained through individual Markov processes, the RBD is equivalent to a global Markov process modelling the whole system. Such a model is called "RBD driven Markov process". This is the basis for most of the probabilistic calculations achieved with RBDs.

## 5.2 Pre-requisite/main assumptions

An RBD is an acyclic directed graph (i.e. no loops or retroactions are modelled in an RBD) which can be drawn by using the basic logical structures presented in Table 3. It is used to model the behaviour of a system on the basis of the following fundamental assumptions:

- a) the system has only two states: working ("success" state, "up" state) or failed ("down" state);
- b) the blocks of an RBD model the components of a system or parts (e.g. groups of components) of a system. Each of them has only two states: working ("success" state, "up" state) or failed ("down" state);
- c) the RBD represents the logic linking the success state of the system to the success states of its components (blocks);
- d) each block behaves independently from the others at all times.

The above assumptions have to be generally fulfilled to apply the analytical calculations (i.e. calculations with formulae) developed in this standard. When they are not fulfilled, the analytical calculations can be replaced by Monte Carlo simulation or other techniques like Markov analysis [2] or Petri nets [3] or the dynamic RBDs described in 12.2 and Annex E.

## 5.3 Limitations

The assumptions presented in 5.2 constitute some limitations but there are other limitations which are less obvious when dealing with time dependent probabilities. In particular, the users of this standard should be aware of the issues introduced by the independency requirements which shall be fulfilled at all times. For example:

- a) sequential events are outside the scope of the Boolean models. Therefore they cannot, in principle, be handled by RBDs. Nevertheless, in simple cases like standby redundancy, it is possible to overcome the problem by considering composite blocks (see Table 3 and 7.5.3) independently of the other blocks;
- b) availability or frequency calculations of repaired systems assume that the repairs of repaired blocks are independent from each other all the time, i.e. each block has its own repair team;

- c) reliability calculations of repaired systems imply that a failed block can be repaired only if the system is still operating when the block failure occurs. This introduces systemic dependencies between the blocks states, and between the blocks and the system states (see 10.3.1.4). This infringes the assumption described in 5.2 d) and so, except in particular cases and with approximations, analytical reliability calculations are generally not possible.

In short, provided that the assumptions in 5.2 are fulfilled, the RBD technique can be used straightforwardly for qualitative analysis and availability/frequency calculations but it can be used for reliability calculation only in particular cases.

It should be noted that, when dealing with probabilistic calculations, good approximations are available with low probabilities (e.g. failure of components/blocks) which cannot be used with high probabilities (e.g. probabilities of success of components/blocks). Therefore to overcome this limitation, it is often better to work with probabilities of failure (unavailability or unreliability) rather than probabilities of success (availability or reliability).

## **6 Establishment of system success/failed states**

### **6.1 General considerations**

A prerequisite for constructing system reliability models is a sound understanding of the ways in which the system and its components can operate. Systems often require more than one success/failure definition. These should be defined and listed. An RBD diagram can be made at different levels: system level, sub-system (module) level or assembly level. When an RBD is made for further analysis (for example for FMEA analysis), a level suitable for such analysis has to be chosen.

In addition, there should be clear statements concerning

- the functions to be performed,
- the performance parameters and permissible limits on such parameters,
- the environmental and operating conditions.

After establishing the system's success/failure definition the next step is to identify logical blocks in order to divide the system as appropriate for the purpose of the reliability analysis. Particular blocks may represent system substructures, which in turn may be represented by other RBDs (system reduction – see 11.2).

For the quantitative evaluation of an RBD, various methods are available. Depending on the type of structure, simple Boolean techniques (see 7) and/or path and cut set analyses (see 8) may be employed. Calculations may be made using analytical methods (e.g. basic component availability methods) or Monte Carlo simulation. An advantage with Monte Carlo simulation is that the probabilities of the events in the RBD do not have to be combined analytically since the simulation itself takes into account whether each block is failed or functional (see 12.2 and Clause F.5).

Since the RBD describes the logical relations needed for the system to function, the block diagram does not necessarily represent the way in which the hardware is physically connected, although an RBD should generally follow, as far as possible, the physical system connections.

### **6.2 Detailed considerations**

#### **6.2.1 System operation**

It may be possible to use a system in more than one functional mode. If separate systems were used for each mode, such modes should be treated independently of other modes, and separate reliability models should be used accordingly. Therefore, when the same system is

used to perform all these functions, separate diagrams should be used for each type of operation. Clear statements of what constitutes system success/failure for each aspect of system operation is a prerequisite.

### 6.2.2 Environmental conditions

The system performance specifications should be accompanied by a description of the environmental conditions under which the system is designed to operate. Also included should be a description of all the conditions to which the system will be subjected during transportation, storage and use.

A particular piece of equipment is often used in more than one environment, for example, on shipboard, in an aircraft or on the ground. When this is so, reliability evaluations may be carried out using the same RBD each time but using the appropriate component/block failure rates for each environment.

### 6.2.3 Duty cycles

The relationship between calendar time, operating time and on/off cycles should be established. If it can be assumed that the process of switching equipment on and off does not in itself promote failures, and that the failure rate of equipment during non-use periods is negligible, then only the actual working time of the equipment needs to be considered.

However, in some instances, the process of switching on and off is in itself the prime cause of equipment failure, and equipment may have a higher failure rate in non-use period than when in-service (e.g. due to moisture and corrosion). In complex cases where only parts of the system are switched on and off, modelling techniques other than RBDs (e.g. Markov analysis or Petri nets) may be more suitable.

## 7 Elementary models

### 7.1 Developing the model

The first step is to select a system success/failure definition. If more than one definition is involved, a separate RBD may be required for each. The next step is to divide the system into blocks to reflect the logical behaviour so that each block is statistically independent of the others. Attempt should be made to make the blocks as large as possible while ensuring that each block contains (preferably) no redundancy.

The next step is to refer to the system success/failure definition and construct a diagram that connects the blocks to form a "success path" (see 3.15). As indicated in the diagrams that follow, the various success paths, between the input and output of the diagram, pass through those combinations of blocks that need to function in order that the system functions.

NOTE In practice, depending on the system configuration, it can be necessary to make repeated attempts at constructing the RBD (each time bearing in mind the steps referred to above) before a suitable block diagram is finalized.

### 7.2 Series structures

If all the blocks are required to function for the system to function, then the corresponding RBD will be one in which all the blocks are connected in series as illustrated in Figure 2.

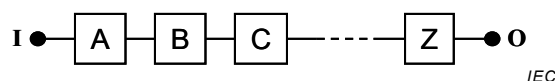


Figure 2 – Series reliability block diagram

In this diagram "I" is the input, "O" the output and A, B, C, ... Z are the blocks which together constitute the system. RBDs of this type are known as "series" RBDs or "series models".

This structure models the following logical function:  $s = a \cdot b \cdot c \cdot \dots \cdot z$  (1)

where  $a$ ,  $b$ ,  $c$  and  $z$  represent the success states of the blocks A, B, C and Z (see Table 2) and  $s$  the success state of the corresponding system.

### 7.3 Parallel structures

A different type of RBD is needed when only one system component (i.e. one block) is required for system success. This is the case when redundant components are implemented.

This is modelled by parallel structures such as that presented in Figure 3 and which represent several redundant blocks. In this structure, the system is down if and only if all blocks are down.

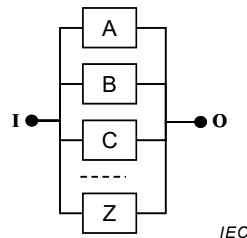


Figure 3 – Parallel reliability block diagram

This structure models the following logical function:  $s = a + b + c + \dots + z$  (2)

### 7.4 Mix of series and parallel structures

The basic structure presented in Figure 2 and Figure 3 can be used to model more complex RBDs. For example, if the entire RBD presented in Figure 2 is duplicated (i.e. made redundant), then the RBD illustrated by Figure 4 is obtained. Alternatively, if each block within the RBD presented in Figure 2 is duplicated, the RBD illustrated by Figure 5 is obtained. Diagrams of this type are known as "series/parallel" RBDs or "series/parallel" models. Note that the terms "duplicated", "redundant" and "parallel" are very similar in meaning but should not be used interchangeably.

- 1) Duplicated is related to the way the RBD is built by repeating similar structures. For example, Figure 4 is the duplication of the structure presented in Figure 2 and Figure 5 is only the duplication of the components. In fact the parallel structures (B1,B2), (C1,C2) etc. are the duplication in series of the parallel structure (A1, A2).
- 2) Redundant is related to the fact that if one component fails, another one can perform its function. For example, A1 and A2 in Figure 5 are redundant.
- 3) Parallel is related to the logic of the architecture of the system and to the graphical representation. For example, A1 and A2 are drawn in parallel in Figure 5 because they are redundant.

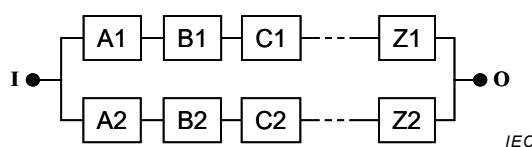
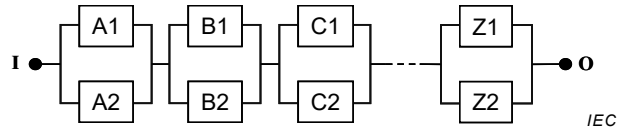


Figure 4 – Parallel structure made of duplicated series sub-RBD

This structure models the following logical function:

$$s = (a_1 \bullet b_1 \bullet \dots \bullet z_1) + (a_2 \bullet b_2 \bullet \dots \bullet z_2) \quad (3)$$

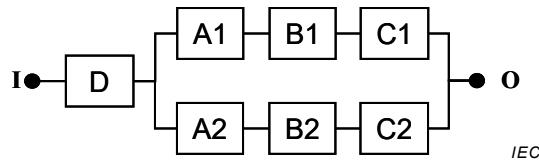


**Figure 5 – Series structure made of parallel reliability block diagram**

This structure models the following logical function:

$$s = (a_1 + a_2) \bullet (b_1 + b_2) \bullet \dots \bullet (z_1 + z_2) \quad (4)$$

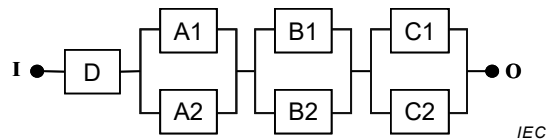
RBDs used for modelling system reliability are often more complicated mixtures of series and parallel structures. For example, a duplicated communication link comprising three repeaters (A1, B1, C1 and A2, B2, C2), and a common power supply block (D) may take the form of, for example, Figure 6 or Figure 7.



**Figure 6 – General series-parallel reliability block diagram**

This structure models the following logical function:

$$s = d \bullet [(a_1 \bullet b_1 \bullet c_1) + (a_2 \bullet b_2 \bullet c_2)] \quad (5)$$



**Figure 7 – Another type of general series-parallel reliability block diagram**

This structure models the following logical function:

$$s = d \bullet (a_1 + a_2) \bullet (b_1 + b_2) \bullet (c_1 + c_2) \quad (6)$$

On account of the assumed statistical independence stated above, failure of any block does not give rise to a change in the probability of failure of any other block within the system. In particular, failure of a redundant block does not affect system power supplies or signal sources.

## 7.5 Other structures

### 7.5.1 *m* out of *n* structures

The need frequently arises to model systems where the success definition is that *m* or more out of *n* items connected in parallel are required for system success. Such logical structures are often called "majority vote" structures. For example see RBDs shown in Figure 8 or Figure 9.

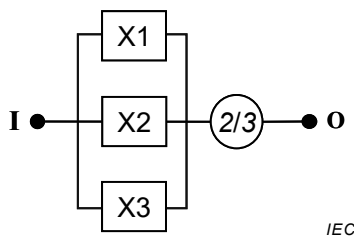


Figure 8 – 2 out of 3 redundancy

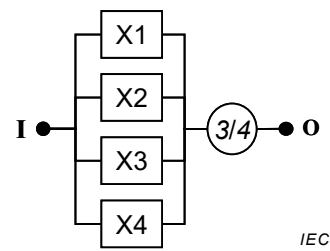


Figure 9 – 3 out of 4 redundancy

Thus, in Figure 8 at least 2 blocks are required for system success and in Figure 9, 3 blocks are required for system success. In both cases the failure of one item is tolerated but failure of two or more items is not.

These structures model the following logical functions:

$$- \text{2/3 redundancy: } s = x_1 \cdot x_2 + x_1 \cdot x_3 + x_2 \cdot x_3 ; \quad (7)$$

$$- \text{3/4 redundancy: } s = x_1 \cdot x_2 \cdot x_3 + x_1 \cdot x_2 \cdot x_4 + x_1 \cdot x_3 \cdot x_4 + x_2 \cdot x_3 \cdot x_4 . \quad (8)$$

These logical functions cannot be represented by a simple combination of elementary series and parallel structure without the implementation of repeated blocks.

### 7.5.2 Structures with common blocks

Most RBDs are easily understood and the conditions for system success are evident. Not all RBDs, however, can be simplified to combinations of series or parallel structures with blocks appearing only once. The RBD in Figure 10 is an example with a block A being common to two paths.

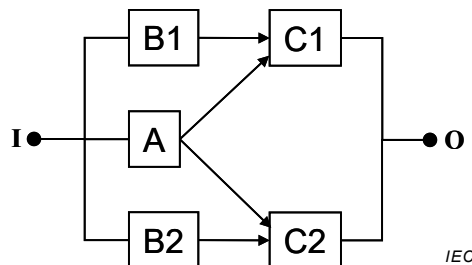


Figure 10 – Diagram not easily represented by series/parallel arrangement of blocks

This structure models the following logical function:

$$s = (b_1 \cdot c_1) + a \cdot (c_1 + c_2) + (b_2 \cdot c_2) \quad (9)$$

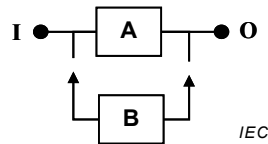
Again, the diagram is self-explanatory. System success is achieved if blocks B1 and C1 are both in up state, or blocks A and C1, or A and C2, or finally B2 and C2. Figure 10 could represent the fuel supply to engines of a light aircraft. B1 represents the supply to the port engine (C1), B2 represents the supply to the starboard engine (C2), and A represents a common backup supply to both engines. The system success definition is that at least one engine needs to be working for aircraft success, or alternatively, both engines need to fail for the aircraft to fail.

It should be noted that in all the above diagrams (Figure 2 to Figure 10), no block appears more than once in a given diagram. The procedures for developing the reliability expression



for diagrams of this type are outlined in 8.2. Figure 18 and Figure 19 provide series-parallel RBDs equivalent to Figure 10 where repeated blocks are implemented.

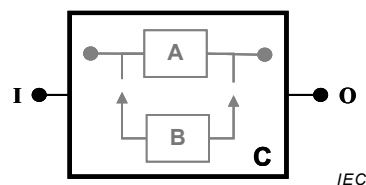
### 7.5.3 Composite blocks



**Figure 11 – Example of RBD implementing dependent blocks**

Figure 11 models a system with cold standby redundancy where the item B starts when the item A fails and with a perfect switching from A to B. Then in the corresponding RBD, the blocks A and B are not independent and this structure infringes the fundamental assumption of independency between blocks which is the basis of this standard.

As blocks A and B cannot be considered independently from each other, it is necessary to consider them as a whole and this can be done by the use of a composite block like the block C presented in Figure 12.

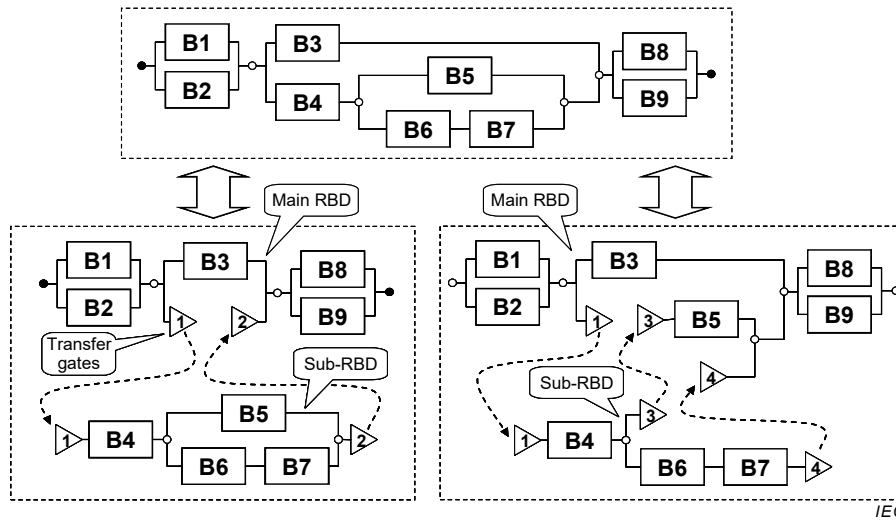


**Figure 12 – Example of a composite block**

The composite block C has two states, success/failure. Then if it is independent from the other blocks of the RBD, it can be handled as a single block. Of course its probability of failure/success has to be calculated by taking into account the blocks A and B and the dependency between them.

### 7.6 Large RBDs and use of transfer gates

RBDs related to industrial systems can be too large to be drawn as a whole on a single sheet of paper. In this case, they can be split in several smaller parts (sub-RBDs) linked together by using transfer gates.



**Figure 13 – Use of transfer gates and sub-RBDs**

Figure 13 gives two examples of the use of transfer gates: each of the two RBDs at the bottom of the figure is equivalent to the RBD at the top. They are split in two parts: main RBDs and sub-RBDs. It should be noted that a sub-RBD does not necessarily need to have only one input and one output.

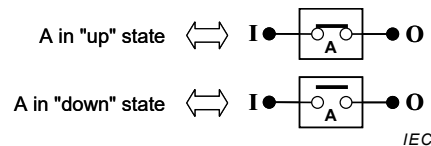
The overall underlying logical function is not affected by such a splitting but this allows the drawing of a large RBD on several separate pages. It is a matter for the analyst to choose a subdivision while keeping a good understanding of the whole RBD and of its sub-RBDs.

## 8 Qualitative analysis: minimal tie sets and minimal cut sets

### 8.1 Electrical analogy

The RBD can be used first for qualitative analysis purposes by identifying

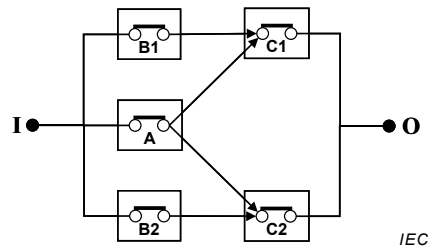
- the combinations of the blocks in up states leading to the system being in the up state ("success" paths or "tie" sets),
- the combinations of the blocks in down states leading to the system being in the down state ("failure" paths or "cut" sets).



**Figure 14 – Analogy between a block and an electrical switch**

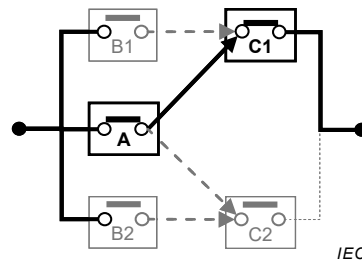
NOTE When building an RBD related to a physical electrical circuit, the position of a physical switch can be different from its representation by using the analogy described in Figure 14. For example, a physical switch stuck closed will be represented by a virtual switch open as it is in down state.

For doing that, the analogy with an electrical circuit shown in Figure 14 is very useful. This consists in considering that each block is equivalent to an electrical switch which is closed when the block is in up state and open when it is in down state. This has been done to represent Figure 10 by the equivalent Figure 15 where each block has been modelled by an electrical switch.



**Figure 15 – Analogy with an electrical circuit**

When this electrical circuit is closed ("tied"), an electrical signal sent from the input circulates throughout the RBD and reaches the output. Therefore, any combination ("set") of closed switches allowing a signal to circulate from the RBD input to the RBD output models an up state of the system. This is called a "success" path with regards to the state of the system or a "tie" set with regards to the closure of the electrical circuit.



**Figure 16 – Example of minimal success path (tie set)**

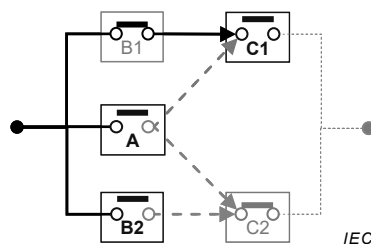
Figure 16 illustrates one of the success paths, (a•c1), of the RBD shown in Figure 15. This success path is minimal as, if A fails or C1 fails, the overall system also fails, i.e. the successes of A and C1 are necessary and sufficient for the system to be in a success state.

B.3.1 gives other examples of minimal and non-minimal tie sets.

The Boolean algebra properties provide a general representation of the system up state,  $s$ , as the union of the minimal tie sets ( $\Pi_i$ ) of the RBD. This leads to the following formula:

$$s = \bigcup_i \Pi_i \quad (10)$$

When this electrical circuit is broken ("cut"), an electrical signal sent from the input is not able to circulate throughout the RBD and does not reach the output. Therefore any combination ("set") of open switches preventing a signal to circulate from the RBD input to the RBD output models a down state of the system. This is called a "failure" path with regards to the state of the system or a "cut" set with regards to the closure of the electrical circuit.



**Figure 17 – Example of minimal failure path (cut set)**

Figure 17 illustrates one of the failure paths,  $(\bar{a} \bullet \bar{b}_2 \bullet \bar{c}_1)$ , of the RBD shown in Figure 15. This failure path (cut set) is minimal as when any of A, B2 or C1 are repaired (switches closing), the system is also repaired, i.e. the failures of A, B2 and C1 are necessary and sufficient for the system being in failed state.

B.3.1 gives others examples of minimal and non-minimal cut sets.

The Boolean algebra properties provide a general representation of the down state,  $\bar{s}$ , as the union of the minimal cut sets ( $C_j$ ) of the RBD:

$$\bar{s} = \bigcup_j C_j \quad (11)$$

Therefore, from Formulae (10) and (11) the identity:  $s = \bigcup_i \Pi_i \equiv \overline{\bigcup_j C_j}$  (12)

is obtained.

The minimal cut sets and minimal tie sets can be obtained by expanding from the logical formulae corresponding to the RBD. Except in simple cases, this is not easy to do that by hand but powerful algorithms are available and implemented into RBD software packages.

## 8.2 Series-parallel representation with minimal success path and cut sets

The identity (12) provides two equivalent ways to represent an RBD from its minimal tie sets or its minimal cut sets.

Applied to the RBD presented in Figure 10, this leads to the two equivalent logical formulae (see detailed explanations in B.3.2):

$$s = \bigcup_i \Pi_i = b_1 \bullet c_1 + a \bullet c_1 + a \bullet c_2 + b_2 \bullet c_2 \quad (13)$$

$$s = \overline{\bigcup_j C_j} = (b_1 + a + b_2) \bullet (c_1 + c_2) \bullet (b_1 + a + c_2) \bullet (b_2 + a + c_1) \quad (14)$$

Then this RBD can be replaced by the equivalent representations presented in Figure 18 (made of tie sets) and in Figure 19 (made of cut sets) where it can be noticed that some blocks are repeated several times.

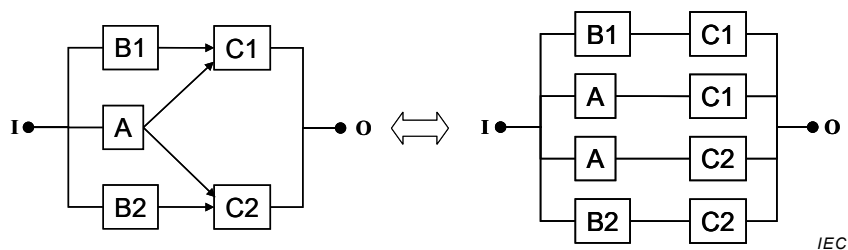
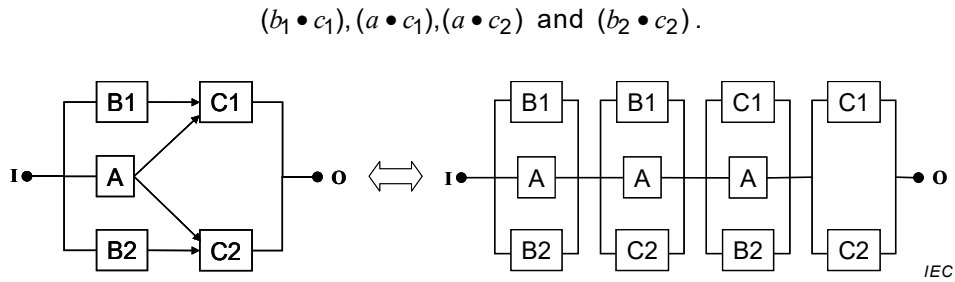


Figure 18 – Equivalent RBDs with minimal success paths

Figure 18 is made of four tie sets of order two (see 3.16, Note 2 to entry):



**Figure 19 – Equivalent RBDs with minimal cut sets**

Figure 19 is made of one cut set of order two ( $\bar{c}_1 \bullet \bar{c}_2$ ) and of three cut sets of order three:  $(\bar{a} \bullet \bar{b}_1 \bullet \bar{b}_2)$ ,  $(\bar{a} \bullet \bar{b}_1 \bullet \bar{c}_2)$  and  $(\bar{a} \bullet \bar{b}_2 \bullet \bar{c}_1)$ .

### 8.3 Qualitative analysis from minimal cut sets

For performing qualitative analysis it is more useful to consider the minimal cut sets rather than the minimal tie sets. This can be shown by using the above example: the cut set of order two ( $\bar{c}_1 \bullet \bar{c}_2$ ) is likely to be more probable than the cut sets of order three  $(\bar{a} \bullet \bar{b}_1 \bullet \bar{b}_2)$ ,  $(\bar{a} \bullet \bar{b}_1 \bullet \bar{c}_2)$  or  $(\bar{a} \bullet \bar{b}_2 \bullet \bar{c}_1)$ . Therefore, from a qualitative point of view, the minimal cut set ( $\bar{c}_1 \bullet \bar{c}_2$ ) is the weak point of the system and should be improved first.

Therefore, the qualitative analysis may be performed with the following steps:

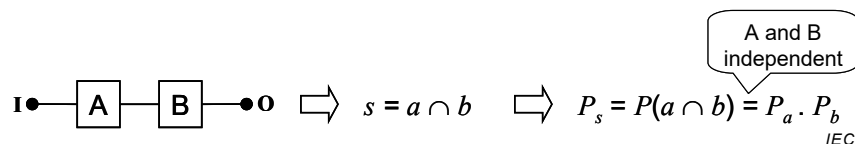
- a) identify the minimal cut sets from the logical equation of the system failure;
- b) sort the minimal cut sets in their increasing orders;
- c) focus on the lowest order minimal cut sets to improve the system.

When failure probabilities are available for the blocks, the minimal cut sets can be more accurately sorted at step b) by calculating the probability of occurrence of each of them.

## 9 Quantitative analysis: blocks with constant probability of failure/success

### 9.1 Series structures

Figure 20 shows the link between the Boolean formulae of a basic series structure and the probabilistic calculations.



**Figure 20 – Link between a basic series structure and probability calculations**

This probabilistic formula is basically established for independent blocks with constant probabilities. It expresses the probability of success of the system  $P_S$  as a function of the individual probabilities of success of block A,  $P_A$ , and block B,  $P_B$ . Therefore, the RBD models can be primarily used for systems comprising independent blocks with constant probability of being in the up state.

At this step it would be irrelevant to talk about reliability, availability or failure frequency of the system as those probabilistic measures are only defined for systems with time dependent behaviour.

The formula established in Figure 20 can be easily extended to be used for systems such as those illustrated by Figure 2 (see B.4.1). When blocks A, B, ..., Z are independent, the probability of success of the system is given by the simple equation:

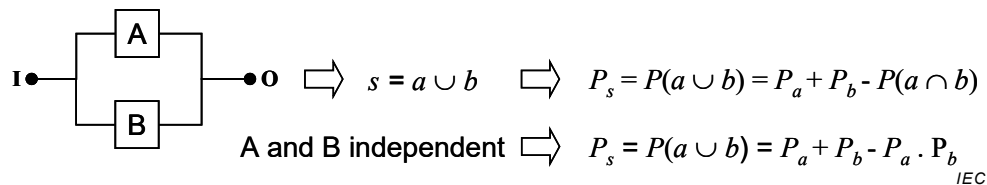
$$P_s = P_a \cdot P_b \cdot P_c \dots P_z \quad (15)$$

i.e. by multiplying together the probabilities of success of all the blocks constituting the RBD.

In general, with  $n$  blocks  $B_i$  in series,  $P_s = \prod_{i=1}^n P_{b_i}$ . (16)

## 9.2 Parallel structures

Figure 21 shows the link between the Boolean formulae of a basic parallel structure and the probabilistic calculations.



**Figure 21 – Link between a parallel structure and probability calculations**

As for the basic series structure, the formula for the basic parallel structure is established for constant probabilities and independent blocks in order to express  $P_s$  as a function of  $P_a$  and  $P_b$ .

As it is, the formula shown in Figure 21 is not easy to extend to more than two components (see the Sylvester-Poincaré formula in 11.7 and B.4.2). Fortunately, it can be observed that  $P_s^c = (1 - P_s) = 1 - (P_a + P_b - P_a \cdot P_b) = (1 - P_a) \cdot (1 - P_b)$ . This expresses simply that the system is failed when both A and B are failed.

Hence the probability of success of the system ( $P_s$ ) is given by:

$$P_s = P_a + P_b - P_a \cdot P_b = 1 - (1 - P_a)(1 - P_b) \quad (17)$$

Formula (17) can be easily extended to  $n$  blocks  $B_i$  in parallel (see B.4.2), i.e.:

– Probability of failure: 
$$P_s^c = \prod_{i=1}^n (1 - P_{b_i}) \quad (18)$$

– Probability of success: 
$$P_s = (1 - P_s^c) = 1 - \prod_{i=1}^n (1 - P_{b_i}) \quad (19)$$

## 9.3 Mix of series and parallel structures

Formulae (15) and (17) can be combined and this can be done by hand in simple cases but the above calculations are generally not easily tractable by hand. Fortunately, powerful

algorithms are available and implemented into RBD software packages. They are based on the techniques described in Clauses B.5, B.6 or B.7.

#### 9.4 $m/n$ architectures (identical items)

The  $m/n$  architectures are analysed in B.4.4. When the blocks are identical (i.e. each with the same probability of success  $p$ ), the probability of success of the system  $P_s$  is given by:

$$P_s = \sum_{r=0}^{n-m} \binom{n}{r} \cdot p^{n-r} \cdot (1-p)^r \quad (20)$$

and the probability of failure is given by:

$$P_f = \sum_{r=0}^{m-1} \binom{n}{r} \cdot (1-p)^{n-r} \cdot p^r \quad (21)$$

When  $n = 2m-1$  (e.g. 1/1, 2/3, 3/5, etc.), the system is up if  $m$  blocks are "up" and the system is down if  $m$  blocks are "down". Those structures are symmetrical with regards to the success and failure events. Some of them, for example structure 2/3, are widely used for safety systems.

If the  $n$  items are not identical, use of a more general procedure is recommended (see 11.8.2).

## 10 Quantitative analysis: blocks with time dependent probabilities of failure/success

### 10.1 General

The calculations developed for constant probabilities in Clause 9 can be easily extended to the time dependent probability of the system  $P_s(t)$  provided that the probabilities  $P_{x_i}(t)$  of the blocks behave independently from each other. This means that the failure (or repair) of any block shall not affect the probability of failure or repair of any other block within the system being modelled. This implies that sufficient repair resources are available to service those blocks needing repair and that, when two or more persons are repairing a particular block at the same time, neither gets in the other's way. Thus failures and repairs of individual blocks are considered to be statistically independent events. The calculations for the time dependent probability of system success are detailed in Annex C.

As the probability for an item to be in the up state at a given instant is its instantaneous availability,  $A_S(t) = P_s(t)$  and  $A_{X_i}(t) = P_{x_i}(t)$ . This result holds for complicated structures as well as large RBDs (see Clause 11 and Annex B): provided that the blocks behave independently from each other at all times, the formulae developed for the constant probability case are still valid for availability/unavailability calculations.

$$- \text{ Series structures: } A_S(t) = \prod_{i=1}^n A_i(t) \text{ and } U_S(t) = 1 - \prod_{i=1}^n [1 - U_i(t)] \quad (22)$$

$$- \text{ Parallel structures: } A_S(t) = 1 - \prod_{i=1}^n (1 - A_i(t)) \text{ and } U_S(t) = \prod_{i=1}^n (U_i(t)) \quad (23)$$

$$- \quad m/n \text{ structure: } A_S(t) = \sum_{r=0}^{n-m} \binom{n}{r} \cdot A(t)^{n-r} \cdot [1-A(t)]^r \quad \text{and} \quad U_S(t) = \sum_{r=0}^{m-1} \binom{n}{r} \cdot U(t)^{n-r} \cdot [1-U(t)]^r \quad (24)$$

The availability/unavailability calculations described above are not so simple but reliability/unreliability calculations are even more difficult. This is due to the definition of the reliability itself:  $R_S(t) = P(\text{S in "up" state over } [0, t])$  which implies that the system remains in up state over the time interval  $[0, t]$ . That means that only the sequences of system events which do not go through to the system down state are relevant for calculating  $R_S(t)$ . Therefore the sequences of system events which include a succession "up state" → "down state" → "up state" have to be excluded from the calculations. This implies that a system part going to the down state is repairable only if the system remains in up state during the repair of this part. Thus, with regards to the calculation of  $R_S(t)$ , the system parts (e.g. components) are repairable or not depending on the system state (i.e. on the states of the other parts). This constitutes a systemic dependency between the system parts and therefore between the blocks modelling these parts within the RBD modelling the system. This happens in the case of redundancy of repairable items. This is the main difficulty in the understanding of this standard: except for RBDs made of blocks in series, the system reliability  $R_S(t)$  cannot be calculated by combining the reliability  $R_{Bi}(t)$  of its individual blocks. The formulae established above under the independency hypothesis are no longer valid. This is further discussed in 10.3.1.4.

## 10.2 Non-repaired blocks

### 10.2.1 General

When a block X is not repaired, its probability to be available at time  $t$  is equal to its probability to have had no failure over  $[0, t]$ . Therefore, its reliability  $R_X(t)$  is equal to its availability  $A_X(t)$ .

When none of the blocks within a system are repaired, the system made of these blocks is not repaired either. Then its availability and reliability are identical and  $R_S(t) = A_S(t)$

### 10.2.2 Simple non-repaired block

The reliability of any item X is linked to its failure rate  $\lambda_X(t)$ , with the following relationship:

$$A_X(t) = R_X(t) = \exp\left(-\int_0^t \lambda_X(u) du\right) \quad (25)$$

where  $\lambda_X(u)$  denotes the failure rate of the block X at  $t = u$ ,  $u$  being a dummy variable.

When  $\lambda_X$  is constant, Figure 25 is simplified to the classical formula:

$$A_X(t) = R_X(t) = \exp(-\lambda_X \cdot t) \quad (26)$$

$$\text{Therefore} \quad U_X(t) = F_X(t) = 1 - \exp(-\lambda_X \cdot t) \quad (27)$$

### 10.2.3 Non-repaired composite blocks

A non-repaired composite block C can be handled as a whole and as a simple non-repaired block provided its availability  $A_C(t)$  is established. Note that in this case  $A_C(t) = R_C(t)$ .

This can be illustrated by the composite block presented in Figure 12. It corresponds to a cold standby system with the following parameters:



- $\lambda_A$  is the constant failure rate of block A and  $f_A(\tau)$  is the probability density function of its time to failure;
- $\lambda_{Bd}$  is the constant failure rate of block B when in a passive (dormant) state, either cold or under low power;
- $\lambda_B$  is the constant failure rate of block B when in an active state, after it has started due to the failure of block A.

NOTE In the following calculations, the switch is considered to be perfect but examples of modelling of the imperfect switching are given in 10.3.1.2 (Figure 23) and C.3.3.

This system is analysed in C.3.3 which provides the following results:

- if the dormant failure rate of item B is assumed to be equal to zero, then the availability of a standby redundant system is:

$$A_C(t) \equiv R_C(t) = e^{-\lambda_A \cdot t} + \frac{\lambda_A}{\lambda_A - \lambda_B} \cdot [e^{-\lambda_B \cdot t} - e^{-\lambda_A \cdot t}] \quad (28)$$

- if both failure rates are equal ( $\lambda_A = \lambda$  and  $\lambda_B = \lambda$ ), then the equation for system reliability can be shown to be given by:

$$A_C(t) \equiv R_C(t) = e^{-\lambda \cdot t} \cdot (1 + \lambda \cdot t) \quad (29)$$

If, under the ideal conditions just above, there are  $n$  (instead of one) items on standby, this latter equation becomes:

$$A_C(t) \equiv R_C(t) = e^{-\lambda \cdot t} \left( 1 + \lambda \cdot t + \frac{(\lambda \cdot t)^2}{2!} + \frac{(\lambda \cdot t)^3}{3!} + \dots + \frac{(\lambda \cdot t)^n}{n!} \right) \quad (30)$$

Formulae (28), (29) or (30) can be used for the composite block C in exactly the same way as Formula (26) is used for ordinary blocks. Nevertheless, establishing those formulae is difficult and other procedures, such as Markov analysis, should be used to analyse standby systems (see 10.3.1.2).

#### 10.2.4 RBDs with non-repaired blocks

- **Availability/reliability:** provided that the blocks behave independently from each other, the availability/reliability of the RBD can be calculated by combining availabilities/reliabilities of the blocks (see 10.2.2 and 10.2.3) according to the logic of the RBD and by using the formulae presented in 10.1.
- **Frequency:** a system made of non-repaired components can fail only once. The probability to observe this failure over  $[0, T]$  is  $F_S(T)$  and the average failure frequency  $w_S^{\text{avg}}(T)$  is equal to  $\frac{F_S(T)}{T}$ . It decreases and tends to zero as time increases.

### 10.3 Repaired blocks

#### 10.3.1 Availability calculations

##### 10.3.1.1 Simple block

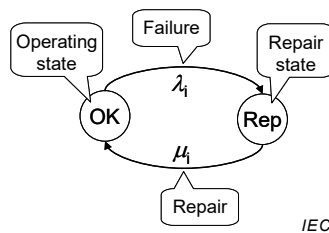
When a block  $i$  is repaired, its availability depends both on its failure rate and on the repair resources. Those resources are generally allocated at the system level and, when they are limited, this constitutes a systemic dependency between the blocks. Therefore, the blocks are independent only if the repair resources are unlimited. In this way the repair of one block can be done at any time even when one or several other blocks are already under repair. This

assumption implies, in particular, that there are as many repair teams as the number of blocks.

The block availabilities  $[A_i(t)]$  can be expressed by any formula (simple or complex). In the simplest case, the repaired blocks are characterized by a constant failure rate  $\lambda_i$  and a constant repair rate  $\mu_i$  and this leads to the classical formula:

$$A_i(t) = \frac{\mu_i}{\lambda_i + \mu_i} + \frac{\lambda_i}{\lambda_i + \mu_i} \exp[-(\lambda_i + \mu_i)t] \tag{31}$$

This analytical formula can be replaced by the equivalent Markov graph presented in Figure 22 where  $A_i(t) = P(\text{OK}, t)$  where  $P(\text{OK}, t)$  is the probability of the state OK at time  $t$ .



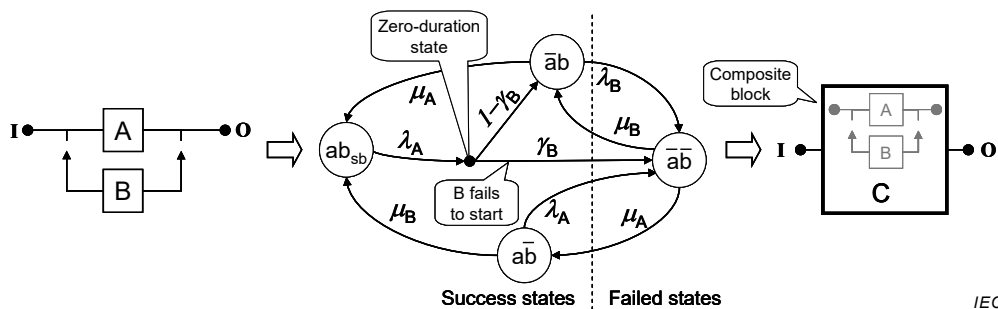
**Figure 22 – "Availability" Markov graph for a simple repaired block**

NOTE A Markov graph devoted to availability calculations is called "availability" Markov graph.

**10.3.1.2 Repaired composite blocks**

A repaired composite block, C, can be handled as a whole and as a simple repaired block provided its availability  $A_C(t)$  is established. Note that in this case  $A_C(t) \neq R_C(t)$ .

This can be illustrated by the composite block presented in Figure 12. It has already been analysed in 10.2.3 in the non-repair case. If the components A and B are considered repairable, then the formula of the availability  $A_C(t)$  of the composite block C can now be established by using the Markov graph in Figure 23.



**Figure 23 – Standby redundancy**

In this graph, C is repaired after it has had a failure (see the transitions from the failed state to the success states): then, this is an "availability" Markov graph (see 10.3.3 to see the difference with a "reliability" Markov graph). This Markov graph can be used to establish availability  $A_C(t)$  of the composite block C or even be used as an RBD input (see C.3).

In this Markov graph, the failure of the switching and sensing mechanism is modelled by using the probability  $\gamma_B$  that block B fails to start when A fails. As this occurs as soon as the component A fails, a zero-duration state has been introduced into the Markov graph. From this state, B immediately starts (probability  $1-\gamma_B$ ) or not (probability  $\gamma_B$ ).

The Markov graph shown in Figure 23 models the dependencies existing between the blocks A and B:

- B starts only after A has failed;
- B may fail when demanded by a failure of A;
- B comes back to a standby position as soon as A and B are in the up state.

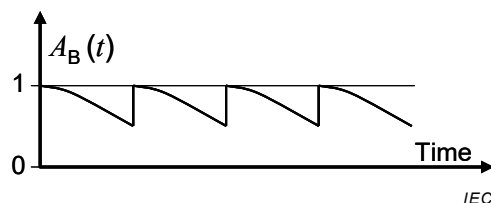
Those dependencies between A and B cannot be taken into account by combining the individual availabilities of A and B and this is why C has to be considered as a whole. If blocks A and B were considered separately, a typical sequential structure outside the scope of the RBD would be obtained. Gathering A and B into a composite block C allows to manage this composite block as an individual block within the RBD framework.

The above principle is general and can be implemented when a few blocks are not independent. When the number of dependent blocks increases, other techniques like dynamic RBDs (see 12.2), Markov processes [2] or Petri nets [3] should be used.

### 10.3.1.3 Periodically tested blocks

With regards to the safety functions that they have to perform, the safety systems are only available or not available. Therefore, they are typical systems with only two states. Their main characteristic is that, in spite of the fact that they remain in the standby position most of the time, they have to react with a high availability when a safety demand occurs.

The components of such safety systems are then periodically tested in order to detect failures that may have occurred when the system is in the standby position. Therefore, the availability of a periodically tested component is maximal just after a test where the possible failures have been detected and repaired, and decreases afterwards until the next test is performed. The typical saw tooth curve shape of the availability  $A_B(t)$  of such a block is illustrated in Figure 24. It can be modelled by a multi-phase Markov process (see Figure C.4). A whole RBD implementing periodically tested blocks is also illustrated in Figure C.5.



**Figure 24 – Typical availability of a periodically tested block**

The shape of the blocks availabilities does not change the principle of the calculation and they can be combined as described above in order to calculate the overall system availability  $A_S(t)$  or unavailability  $U_S(t)$ . This is very useful to implement the average unavailability calculations (i.e.  $PFD_{avg}$ ) required by functional safety standards (e.g. IEC 61508 [5] or IEC 61511 [6]) as explained at the end of 10.3.2.

### 10.3.1.4 Complex repaired blocks (RBD driven Markov processes)

Provided that the independency requirements are fulfilled, the idea developed in Figure 22 and Figure 23 to use small Markov graphs with few states to model the block availabilities can be easily extended to all the blocks of an RBD.

This allows the construction of large Markov models (comprising millions of states) made of small individual sub-Markov models (comprising a few states each) combined through the logic of an RBD. Therefore

- the Markov graphs provide the block availabilities,

– the RBD provides the logic used to combine the block availabilities.

Such models are called "RBD driven Markov processes".

See Clause C.4 for more details.

### 10.3.2 Average availability calculations

Another useful parameter to calculate from an RBD is the average availability  $A_S^{\text{avg}}(0, T)$  of the system over a given period  $[0, T]$ . This can be done by the integration of the instantaneous system availability  $A_S(t)$ :

$$A_S^{\text{avg}}(0, T) = \frac{1}{T} \int_0^T A_S(t) dt \quad (32)$$

In the general case, such calculations are not really possible by hand but nowadays RBD software packages are available to make the needed numerical calculations.

Nevertheless, under certain conditions, a steady state is reached where the probability for  $B_i$  to leave the up state by a failure is equal to the probability for  $B_i$  to reach it by a repair. When a steady state exists, the availability  $A_{B_i}(t)$  reaches an asymptotic value  $A_{B_i}^{\text{as}}$ .

This occurs when

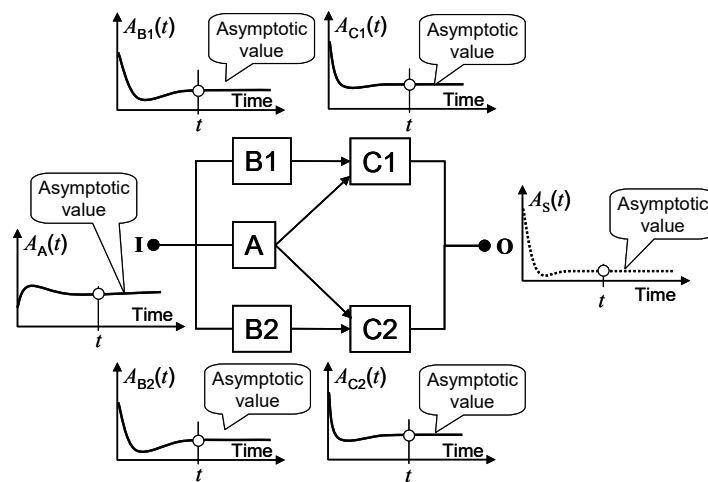
- the failures are quickly detected and repaired (i.e.  $1/\mu_i \ll 1/\lambda_i$ ),
- the failure and repair rates ( $\lambda_i, \mu_i$ ) are constant.

For example, in the case described in Figure 22, the steady state availability of the block is equal to  $A_{B_i}^{\text{as}} = \mu_i / (\lambda_i + \mu_i)$ .

When all the blocks reach such steady states, the system also reaches a steady state (see Figure 25 and Figure C.3) where  $A_S(t) \rightarrow A_S^{\text{as}}$ . Then, outside the transient period,

Equation (32) gives the long term average availability of the system:  $A_S^{\text{avg}} = A_S^{\text{as}}$ .

Therefore, when an RBD reaches a steady state, the steady state availabilities of the blocks become constant and the formulae established in Clause 9 can be used to carry out system steady-state availability predictions. This is accomplished by simply replacing the constant probabilities  $P_{b_i}$  by the constant values  $A_{B_i}^{\text{as}}$ .

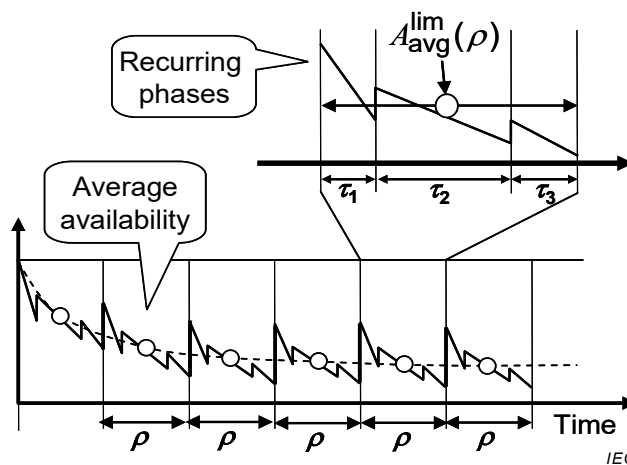


IEC

Figure 25 – Example of RBD reaching a steady state

Warning: the calculations given above are valid only with asymptotic block availabilities. They are not valid with ordinary average availabilities.

Then, when the RBD does not reach a steady state, the average availability has to be calculated by using the general Formula (32).



IEC

Figure 26 – Example of RBD with recurring phases

A special case occurs when the RBD is used through recurring phases such as:

- succession of seasons: winter, spring, summer and autumn;
- test intervals for periodically tested items.

Figure 26 illustrates a system with three recurring phases such that the same pattern of three phases is repeated with a time interval equal to  $\rho = \tau_1 + \tau_2 + \tau_3$ . The availability of such a system has no asymptotical value but the average availability  $A_{avg}[n\rho, (n+1)\rho]$  generally reaches a limit value when  $n$  is large enough:

$$A_{avg}^{Lim}(\rho) = \lim_{n \rightarrow \infty} A_{avg}[n\rho, (n+1)\rho] \tag{33}$$

As  $A_{\text{avg}}[n\rho, (n+1)\rho]$  decreases when  $n$  increases,  $A_{\text{avg}}^{\text{Lim}}(\rho)$  provides a good conservative approximation of the average availability over a time interval  $[0, T]$  encompassing several sets of recurring phases.

Therefore, the techniques described in this standard can be used to calculate the average unavailability of safety systems and this makes the link with the functional safety standards (IEC 61508 or IEC 61511) which require such calculations for safety instrumented systems and where the average unavailability is called  $\text{PFD}_{\text{avg}}$  (average of the probability of failure on demand). This is described in Clauses C.4 and C.5.

### 10.3.3 Reliability calculations

When repaired blocks are considered, the calculation of the reliability  $R_S(t)$  implies that the repairs of the blocks in the system (RBD) need to be considered only as long as the system remains in the up state (see in 10.1).

This can be illustrated by the simple redundant system modelled by the RBD on the left hand side of Figure 27. With regards to the calculation of  $R_S(t)$ , when the block B fails, it can be repaired only if S in the up state (i.e. if A is in the up state). In the same way when the block A fails, it can be repaired only if S in the up state (i.e. if B is in the up state). Therefore when one block fails, its repair depends on the state of the system S which, in turn, depends on the states of all the blocks. This systemic dependency between the blocks is modelled in the Markov graph presented on the right hand side of Figure 27. It is equivalent to the RBD presented on the left hand side.

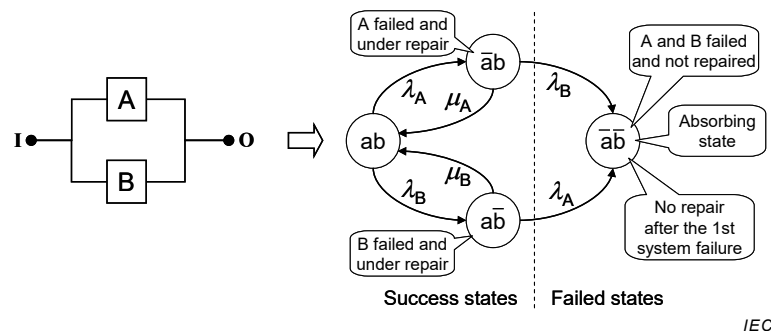


Figure 27 – RBD and equivalent Markov graph for reliability calculations

The simple redundant system is made of two redundant blocks A and B and it has 4 states. The success states are  $ab$ ,  $\bar{a}\bar{b}$  and  $a\bar{b}$  and the failed state is  $\bar{a}\bar{b}$ . The system is reliable over a given period  $[0, t]$  only if it remains in the "success" states all the time. Therefore, the states  $ab$ ,  $\bar{a}\bar{b}$  and  $a\bar{b}$  are "reliable" states only if they come from transitions between each other. That means that, for reliability calculations at time  $t$ , the transition out of  $\bar{a}\bar{b}$  is not allowed during  $[0, t]$ . The state  $\bar{a}\bar{b}$  is an absorbing state and the presence of an absorbing state characterizes a reliability Markov graph.

In this graph, block A can be repaired in the state  $\bar{a}\bar{b}$  but not in the state  $\bar{a}\bar{b}$  and block B can be repaired in the state  $a\bar{b}$  but not in the state  $\bar{a}\bar{b}$ . Therefore, the repair of a failure of a block depends on the state of the whole system when it occurs: this is what is called a "systemic" dependency.

It is no longer possible to calculate the system reliability by combining the individual probabilities of success of the blocks.

- The block availabilities,  $[A_i(t)]$ , cannot be used as this would give the system availability and not the system reliability.

- The block reliabilities,  $[R_i(t)]$ , cannot be used as this would give the reliability of a system with non-repaired blocks (because the reliability of a repaired component is the same as the reliability of a non-repaired component with the same failure rate).

Therefore, except in the particular case developed hereafter, other techniques like Monte Carlo simulation (e.g. DRBDs, see 12.2 and Annex E), Markov [2] or Petri Nets [3] should be used instead.

The only case where reliability calculations are manageable occurs for quickly (i.e.  $MTTR_i \ll MTTF_i$ ) and completely (i.e. every failure is repaired) repaired systems. That means that, when a block fails, the repair starts at once and lasts a short time. In this case the system reaches a steady state rather quickly and its availability  $A_S(t)$  an asymptotic value  $A_S^{as}$ . In this steady state the conditional failure intensity,  $\Lambda_{VS}$  (also called Vesely failure rate) is constant and provides a good approximation of the system failure rate,  $\Lambda_S$ . Then the system reliability is obtained by using the classical formula  $R_S(t) = e^{-\Lambda_S t} \approx e^{-\Lambda_{VS} t}$  and the system unreliability is given by  $F_S(t) = 1 - e^{-\Lambda_S t} \approx 1 - e^{-\Lambda_{VS} t}$ .

For example in Figure 27, when the steady state is reached, the properties of the Markov processes allow to obtain a good approximation of  $\Lambda_{VS}$  and  $\Lambda_S$  directly from the Markov process:

$$\Lambda_S \approx \Lambda_{VS} \approx \lambda_A \frac{\lambda_B}{\lambda_B + \mu_A} + \lambda_B \frac{\lambda_A}{\lambda_A + \mu_B} \quad (34)$$

This is not a simple formula even though the system is very simple. For larger RBDs, the Vesely failure rate,  $\Lambda_{VS}$ , can be obtained from conditional probabilities which can be calculated by the algorithms currently in use in RBD software packages. The way to obtain the conditional and unconditional failure intensities from an RBD is detailed in Clause C.6 and more details are given about reliability calculations in Clause C.7.

### 10.3.4 Frequency calculations

When the blocks are repaired, another useful probabilistic measure is the average failure frequency of the system over a given time interval  $[0, T]$  which is equal to  $n/T$  if  $n$  failures occur over  $T$ . This average failure frequency is obtained by calculating the average unconditional failure intensity of the system  $w_S^{avg}(0, T) = n/T$ .

The failure frequency can be calculated in any case but this is difficult by hand. Algorithms have been developed to do that and the principle is explained in Clause C.6.

## 11 Boolean techniques for quantitative analysis of large models

### 11.1 General

It is possible to evaluate the availability  $A_S(t)$  of all the systems considered so far by the application of a suitable availability formula selected from Formulae (15) to (24). However, when the number of blocks increases, the corresponding RBDs may not conveniently be evaluated by any of the above formulae. Calculations are more difficult and so other mathematical approaches have to be employed.

Such approaches provide several ways to manipulate the Boolean equations in order to make the calculations possible. They can be generally employed manually on small RBDs but most of them can be used through a software package when the number of blocks is large. They are based on the following techniques:

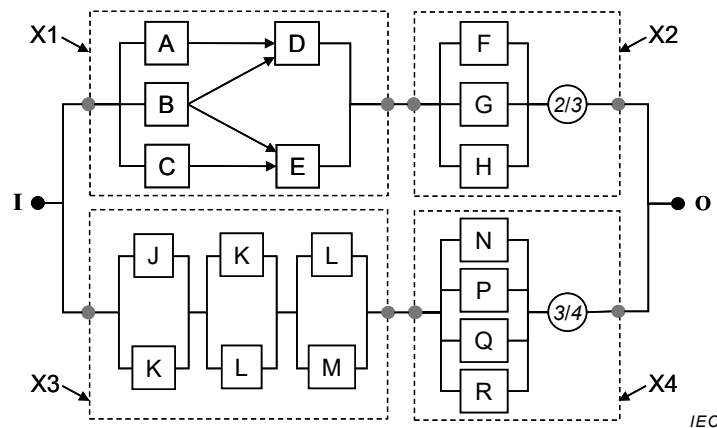
- reduction of the RBD to simpler structures;
- use of the total probability theorem;
- use of Boolean truth tables;
- use of Karnaugh maps;
- use of Shannon decomposition and binary decision diagrams;
- use of general Sylvester-Poincaré formulae.

For the procedures that follow, the condition of independence, as stated in 5.2 d), applies and the formulae provided hereafter for constant probability calculations may be straightforwardly transformed for availability calculations by applying 10.1 and Annex C.

It should be noted that Monte Carlo simulation can also be used for complex RBDs. The use of such procedures is not dealt with in this standard but the dynamic RBDs are described in 12.2 and Annex E.

**11.2 Method of RBD reduction**

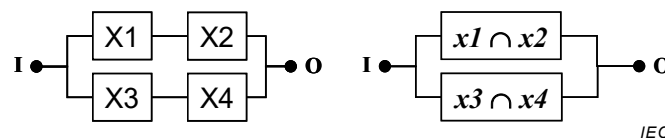
The RBDs modelling an industrial system may seem very complicated. By careful examination, however, the blocks in the diagram can often be grouped together such that the groups are statistically independent. In particular, this means that no two (or more) groups can contain the same block.



**Figure 28 – Illustrating grouping of blocks before reduction**

This can be illustrated by considering the RBD shown in Figure 28.

Figure 28 can be reduced to the diagram shown in Figure 29 which is made of the four dotted groups of blocks X1, X2, X3 and X4 as illustrated in Figure 10, Figure 8, Figure 37, and Figure 9 respectively.



**Figure 29 – Reduced reliability block diagrams**

Hence the final system availability is given by

$$A_S(t) = A_{X1} \cdot A_{X2} + A_{X3} \cdot A_{X4} - A_{X1} \cdot A_{X2} \cdot A_{X3} \cdot A_{X4} \tag{35}$$



as explained in 9.2.

This technique of reduction is difficult to automate but very useful for calculations by hand.

### 11.3 Use of total probability theorem

When dealing with RBDs of the type illustrated by Figure 10 which implement a common block A, it is possible to implement an approach based on the total probability theorem.

Two mutually exclusive events  $x$  and  $\bar{x}$  form a complete set of events (i.e.  $x + \bar{x} = \Omega$ ) and the total probability theorem can be summarized as follows:

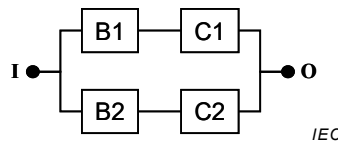
$$P_s = P_{s|x} \cdot P_x + P_{s|\bar{x}} \cdot P_{\bar{x}} = P_{s|x} \cdot P_x + P_{s|\bar{x}} \cdot (1 - P_x) \quad (36)$$

In Equation (36)  $P_s$  denotes the probability of success of a system,  $P_{s|x}$  denotes the probability of success of the system given that a particular item X is working, and  $P_{s|\bar{x}}$  denotes the probability success of the system given that the particular item X has failed.

Formula (36) can be applied to the block A of Figure 10 and this leads to:

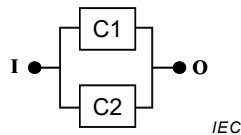
$$P_s = P_{s|a} \cdot P_a + P_{s|\bar{a}} \cdot P_{\bar{a}} \quad (37)$$

For example, when the item A has failed, the RBD of Figure 10 becomes the RBD shown in Figure 30 so that  $P_{s|\bar{a}} = P_{b1} \cdot P_{c1} + P_{b2} \cdot P_{c2} - P_{b1} \cdot P_{c1} \cdot P_{b2} \cdot P_{c2}$



**Figure 30 – Representation of Figure 10 when item A has failed**

Similarly, when A is working, the RBD of Figure 10 becomes that given in Figure 31 so that  $P_{s|a} = P_{c1} + P_{c2} - P_{c1} \cdot P_{c2}$ .



**Figure 31 – Representation of Figure 10 when item A is working**

$$\text{Hence } P_s = (P_{c1} + P_{c2} - P_{c1} \cdot P_{c2}) \cdot P_a + (P_{b1} \cdot P_{c1} + P_{b2} \cdot P_{c2} - P_{b1} \cdot P_{c1} \cdot P_{b2} \cdot P_{c2}) \cdot (1 - P_a) \quad (38)$$

If  $P_{c1} = P_{c2} = P_c$  and  $P_{b1} = P_{b2} = P_b$ , the above Formula (38) simplifies to:

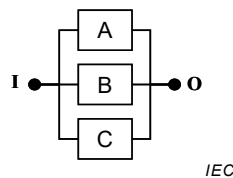
$$P_s = (2P_c - P_c^2) \cdot P_a + (2P_b \cdot P_c - P_b^2 \cdot P_c^2) \cdot (1 - P_a) \quad (39)$$

This procedure can be extended to  $n$  mutually exclusive events  $a_1, \dots, a_n$ , whose probabilities sum to unity (i.e.,  $a_1 + a_2 + \dots + a_n = \Omega$ ), then  $P_s = P_s | a_1 \cdot P_{a_1} + \dots + P_s | a_n \cdot P_{a_n}$ . It can be used to deal with an RBD with repeated blocks (see 11.8.1.2). When there are  $n$  repeated blocks, this leads to develop  $2^n$  terms for the formula of  $P_s$ . Therefore, this technique is useful to deal with RBDs with a limited number of repeated blocks for example  $m/n$  structures.

**11.4 Use of Boolean truth tables**

The system success paths depicted by RBDs are the graphical description of an underlying Boolean expression. For example, three redundant items A, B and C (one out of three required for system success) can be represented by the parallel RBD configuration illustrated in Figure 32, or by the Boolean expression:

$$s = a + b + c \tag{40}$$



**Figure 32 – RBD representing three redundant items**

The Sylvester-Poincaré formula (see 11.7 and B.5) applied to three independent events leads to the following result:

$$P_s = P_a + P_b + P_c - (P_a \cdot P_b + P_a \cdot P_c + P_b \cdot P_c) + P_a \cdot P_b \cdot P_c \tag{41}$$

Formula (41) comprises seven terms when there are only three blocks (i.e. three events  $a$ ,  $b$  and  $c$ ). This number of terms increases exponentially when the number of involved events increases.

To prevent the increasing of terms, the idea is then to replace the events  $a$ ,  $b$  and  $c$  by equivalent combinations of disjoint events (see 11.7) and this can be done by using the truth table of the system according to the states of blocks A, B and C.

**Table 5 – Application of truth table to the example of Figure 32**

State number	Block			System	Disjointed terms	Consensus	
	A	B	C				
1	0	0	0	0	$\bar{a} \cdot \bar{b} \cdot \bar{c}$		
2	0	0	1	1	$\bar{a} \cdot \bar{b} \cdot c$	$\bar{a} \cdot \bar{b} \cdot c$	$\bar{a} \cdot \bar{b} \cdot c$
3	0	1	0	1	$\bar{a} \cdot b \cdot \bar{c}$	$\bar{a} \cdot b$	$\bar{a} \cdot b$
4	0	1	1	1	$\bar{a} \cdot b \cdot c$		
5	1	0	0	1	$a \cdot \bar{b} \cdot \bar{c}$	$a \cdot \bar{b}$	a
6	1	0	1	1	$a \cdot \bar{b} \cdot c$		
7	1	1	0	1	$a \cdot b \cdot \bar{c}$	$a \cdot b$	
8	1	1	1	1	$a \cdot b \cdot c$		

NOTE 1= working, 0 = failed.

This truth table identifies 8 disjoint terms representing the 8 possible states of the system: the state number 1 corresponds to the failure of the system and the states 2 to 8 to the success states of the system.

Again there are seven terms to handle and therefore this raw decomposition in disjointed terms is not very effective to calculate  $P_s$ . Note that there is generally no link between the number of terms of the Sylvester-Poincaré formula and the number of the disjoint terms from the truth table.

Fortunately, some disjoint terms can be combined through "consensus" as this has been done on the right hand side of the table. In a first step, terms 7 and 8, 5 and 6 and 3 and 4 have been combined together to obtain 4 disjoint terms. In a second step, terms 5 to 7 have been merged into a single term and three disjoint terms are obtained:

$$s = (a + b + c) \equiv a + (\bar{a} \cdot b) + (\bar{a} \cdot \bar{b} \cdot c) \quad (42)$$

Finally, the probability of failure of the system can be calculated as:

$$P_s = P_a + (1 - P_a) \cdot P_b + (1 - P_a) \cdot (1 - P_b) P_c \quad (43)$$

Formula (43) can be directly used to evaluate the system availability:

$$A_S(t) = A_A(t) + [1 - A_A(t)] \cdot A_B(t) + [1 - A_A(t)] \cdot [1 - A_B(t)] \cdot A_C(t) \quad (44)$$

It has to be noted that Table 5 identifies only one term leading to the system down state  $\bar{a} \cdot \bar{b} \cdot \bar{c}$ . Therefore, the calculation of the probability of failure is simpler than the probability of success and:

$$P_s = 1 - P_{\bar{s}} = 1 - P_{\bar{a}} \cdot P_{\bar{b}} \cdot P_{\bar{c}} \quad (45)$$

and finally 
$$A_S(t) = 1 - U_S(t) = 1 - U_A(t) \cdot U_B(t) \cdot U_C(t) \quad (46)$$

Formulae (44) and (46) are equivalent.

For a system with  $n$  blocks, the Boolean truth table has  $2^n$  rows and therefore this approach can soon become unwieldy, although the principle involved is quite straightforward. This problem is overcome to some extent by using Karnaugh maps (see 11.5) but it is actually solved by using the Shannon decomposition and the binary decision diagrams explained hereafter (see 11.6). For a detailed description of a general application of Boolean methods, see Annex B.

### 11.5 Use of Karnaugh maps

The Karnaugh maps [8][9][10][11] technique has been developed to simplify the logical equation corresponding to a truth table. Therefore, it can be used for RBDs as well.

The principle of the use of such maps is illustrated hereafter with maps related to the RBD presented in Figure 10. This RBD comprises 5 blocks. As the Karnaugh maps are easier to manipulate with 4 variables, the whole map has been split into two disjoint cases:

- A is in up state (Table 6);
- A is in down state (Table 7).

**Table 6 – Karnaugh map related to Figure 10 when A is in up state**

B1 B2 \ C1 C2	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

The Karnaugh map in Table 6 shows a 4x4 grid. The top row is labeled '00' and the bottom row '10'. The left column is labeled '00' and the right column '10'. The middle two columns are labeled '01' and '11'. A callout 'c2' points to the rightmost column (11 and 10). A callout 'c1' points to the bottom two rows (11 and 10). A dotted box encloses the bottom two rows (11 and 10) across all columns. A solid box encloses the middle two rows (01 and 11) across all columns.

**Table 7 – Karnaugh map related to Figure 10 when A is in down state**

B1 B2 \ C1 C2	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	1	1	1
10	0	0	1	1

The Karnaugh map in Table 7 shows a 4x4 grid. The top row is labeled '00' and the bottom row '10'. The left column is labeled '00' and the right column '10'. The middle two columns are labeled '01' and '11'. A callout 'b2 • c2' points to the top-right cell (00, 10). A callout 'b1 • c1' points to the bottom-right cell (10, 10). A dotted box encloses the bottom two rows (11 and 10) across all columns. A solid box encloses the middle two rows (01 and 11) across all columns.

In Table 6 and Table 7, the blocks have been split in two groups (B1 B2 and C1 C2) and the states of the components have been organized in such a way that only one state changes from a column to the next one and from a line to the next one. Therefore, the combinations which can be simplified are close together. For example, in Table 6, the combinations boxed in plain lines represent only C2 as the states of C1, B1 and B2 do not matter. In the same way, the combinations boxed in dotted lines represent only C1 as the states of C2, B1 and B2 do not matter. This leads to

$$s|a = c_1 + c_2 \quad (47)$$

where  $s|a$  represents the system S being in up state given the block A is in the up state and  $c_1$  and  $c_2$  represent the state variables related to blocks C1 and C2.

With the Karnaugh map presented in Table 7 the following formula is obtained

$$s|\bar{a} = b_1 \cdot c_1 + b_2 \cdot c_2 \quad (48)$$

where  $s|\bar{a}$  represents the system S being in up state given A is in the down state and  $b_1$ ,  $b_2$ ,  $c_1$  and  $c_2$  represent the state variables related to blocks B1, B2, C1 and C2.

Gathering the results gives:

$$s = a \cdot (c_1 + c_2) + \bar{a} \cdot (b_1 \cdot c_1 + b_2 \cdot c_2) \quad (49)$$

And finally, the system availability can be calculated with the following formula:

$$A_S(t) = A_A(t) \cdot \{1 - [1 - A_{C1}(t)] \cdot (1 - A_{C2})\} + [1 - A_A(t)] \cdot \{1 - [1 - A_{B1}(t) \cdot A_{C1}(t)] \cdot [1 - A_{B2}(t) \cdot A_{C2}(t)]\} \quad (50)$$

The Karnaugh maps have the same number of terms ( $2^n$ ) as the original truth tables but they are more compact and the combinations are organized in a better way to identify the combinations which can be merged together. This is very useful to identify the minimal tie or cut sets.

### 11.6 Use of the Shannon decomposition and binary decision diagrams

Like the truth or Karnaugh maps, the Shannon decomposition allows to identify the disjoint terms of a Boolean equation.

Figure 33 illustrates the principle of the Shannon decomposition on the Boolean function (1 out of 3) related to the RBD presented in Figure 32.

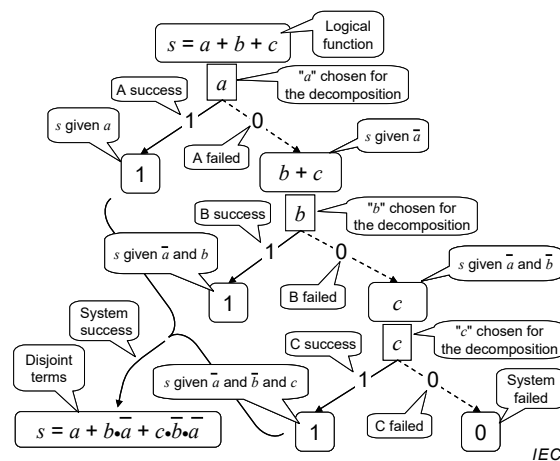


Figure 33 – Shannon decomposition equivalent to Table 5

This decomposition is done in several steps:

- 1) choose the order of the variable appearing in the logical function (here order  $a, b, c$ );
- 2) for each variable draw two branches (success and failure);
- 3) if a state of the system (success or failure) is reached, then stop the decomposition, otherwise continue with the next variable;
- 4) identify the paths leading to the success (or failed) state of the system.

Figure 33 leads to 3 disjoint success paths:  $a, (\bar{a} \cdot b), (\bar{a} \cdot \bar{b} \cdot c)$ . This is the same result as the truth table but it has been obtained in a simpler way. This decomposition is not unique and depends on the order which has been chosen for the variables.

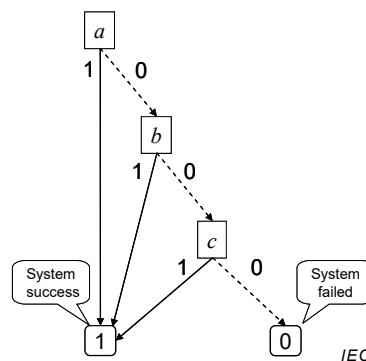


Figure 34 – Binary decision diagram equivalent to Table 5

When the up and down states are gathered, the Shannon decomposition provides the binary decision diagram (BDD) shown in Figure 34. The BDDs provide very compact representations of Boolean equations in the form of disjoint terms. They are very effective in computation and constitute the present state of the art (see references [31], [32] and [33]) for probabilistic calculations on Boolean models (e.g. RBD and fault trees).

### 11.7 Use of Sylvester-Poincaré formula

When the number of components increases, the simple formulae and the manual application of the above techniques become unmanageable because of the combinatorial explosion of the number of terms involved in the calculation.

As this has been explained in Clause 8, an RBD can be represented by the union of its success paths (minimal tie sets).

Then, the probabilistic calculations can be performed from the unions of the tie sets by using the Sylvester-Poincaré formula (see B.4.2 and B.5.2) which is the generalization of the basic formula  $P(a + b) = P_a + P_b - P_a \cdot P_b$ :

$$P_s = P\left(\bigcup_{i=1}^n \Pi_i\right) = \sum_i P(\Pi_i) - \sum_{i < j} P(\Pi_i \cdot \Pi_j) + \sum_{i < j < k} P(\Pi_i \cdot \Pi_j \cdot \Pi_k) - \dots \quad (51)$$

A similar calculation (see B.5.3) may be done by using the failure paths (minimal cut sets):

$$P_s = 1 - P_f = P\left(\bigcup_{i=1}^m C_i\right) = \sum_i P(C_i) - \sum_{i < j} P(C_i \cdot C_j) + \sum_{i < j < k} P(C_i \cdot C_j \cdot C_k) - \dots \quad (52)$$

The Sylvester-Poincaré formula is an alternate sum whose result converges toward the exact value when the number of considered terms increases. The difference is that Formula (51) which handles probabilities  $P(\Pi_i)$  close to 1 converges very slowly when Formula (52) which handles probabilities  $P(C_i) \ll 1$  converges rather quickly. In this case, the first term of Formula (52) provides a conservative estimation of the probability of failure:

$$P_s = P\left(\bigcup_{i=1}^m C_i\right) \approx \sum_i P_{C_i} \quad (53)$$

This approximation is widely used and works well when the probabilities of failures of the blocks are small which is generally true for the components in safety systems. It is the basis of calculations performed by numerous software packages available for availability/reliability calculations on RBD or fault trees.

Nevertheless, it is possible to overcome the difficulty of using Formula (51) by transforming the tie sets into equivalent sets of disjoint terms  $\bigcup_{i=1}^q \Pi_i^d = \bigcup_{i=1}^n \Pi_i$  such as  $\Pi_i^d \cdot \Pi_j^d = \Phi, \forall i, j$ .

In this case, the Sylvester-Poincaré Formula (51) is reduced to its first term:

$$P_s = P\left(\bigcup_{i=1}^q \Pi_i^d\right) = \sum_i P(\Pi_i^d) \quad (54)$$

The same can be done with Formula (52) by replacing the minimal cut sets ( $C_i$ ) by an equivalent set of disjoint terms ( $C_i^d$ ). Then the Sylvester-Poincaré Formula (52) is reduced to its first term:

$$P_s = P\left(\bigcup_{i=1}^m C_i^d\right) = \sum_i P(C_i^d) \tag{55}$$

The disjoint sets can be found by using the truth tables (11.4), the Karnaugh maps (11.5) or the Shannon decomposition and the binary decision diagrams (11.6). At the present time the state of the art to identify the disjoint terms of a Boolean equation is based on the binary decision diagrams (BDDs) described in 11.6. This provides powerful algorithms able to handle very large RBDs comprising a lot of blocks repeated or not repeated.

### 11.8 Examples of RBD application

#### 11.8.1 Models with repeated blocks

##### 11.8.1.1 Cut and tie set representation

In Clause 7 no block in the RBD appeared more than once. It may sometimes be advantageous to use block diagrams of the type illustrated by Figure 35.

The left hand side of Figure 35 shows a conventional RBD with 4 blocks: blocks C and D might model two functionally similar items acting as duplicates for one another, but item A can power only item C, whereas item B is capable of supplying power to both C and D.

The middle and the right hand side of Figure 35 provide 2 equivalent RBDs for modelling not only the physical arrangements of the items, but the RBD as well. It is important for the RBD on the right hand side to include arrows in order to remove the uncertainty which occurs with such a diagram.

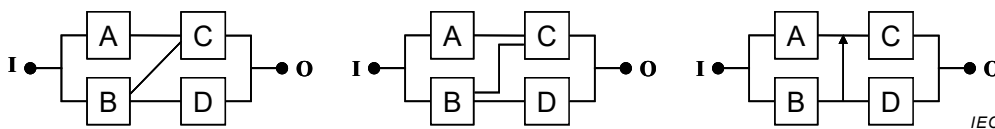


Figure 35 – RBD using an arrow to help define system success

The system success paths  $a \bullet c$ ,  $b \bullet c$ ,  $b \bullet d$  of the system modelled in Figure 35 can be used to build an equivalent RBD in which some blocks appear more than once. When all success paths fail, this would cause the system to fail. Therefore, the RBD can be represented by a parallel combination of such success paths. This is illustrated in Figure 36.

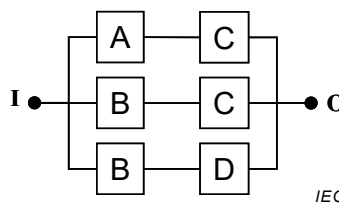
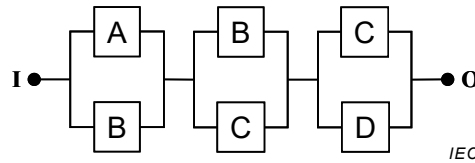


Figure 36 – Alternative representation of Figure 35 using repeated blocks and success paths

Alternatively, the failure paths (e.g., the minimal cut sets of the system)  $\bar{a} \bullet \bar{b}$ ,  $\bar{b} \bullet \bar{c}$ ,  $\bar{c} \bullet \bar{d}$  can be used to build an equivalent RBD. This is done in Figure 37. When all components fail

in one of the minimal cut sets this would cause the system to fail and Figure 37 is thus a series combination of such minimal cut sets.



**Figure 37 – Other alternative representation of Figure 35 using repeated blocks and minimal cut sets**

The representations in Figure 36 and Figure 37 illustrate the concepts covered in 8.2: any RBD can be represented by the parallel combination of its success paths or by the series combination of its minimal cut sets.

Blocks B and C are repeated in both the RBDs presented in Figure 36 and Figure 37. It would be incorrect to treat the blocks as if they were independent of the others. Instead, the methods given in 11.3, 11.5 and 11.6 can be applied.

#### 11.8.1.2 Total probability theorem implementation

The method of the total probability decomposition described in 11.3 applied to Figure 36 and extended for two repeated components gives:

$$\begin{aligned} P_s &= P_{s|b,c} \cdot P_b \cdot P_c + P_{s|b,\bar{c}} \cdot P_b \cdot P_{\bar{c}} + P_{s|\bar{b},c} \cdot P_{\bar{b}} \cdot P_a + P_{s|\bar{b},\bar{c}} \cdot P_{\bar{b}} \cdot P_{\bar{c}} \\ &= P_{s|b,c} \cdot P_b \cdot P_c + P_{s|b,\bar{c}} \cdot P_b \cdot (1 - P_c) + P_{s|\bar{b},c} \cdot (1 - P_b) \cdot P_a + P_{s|\bar{b},\bar{c}} \cdot (1 - P_b) \cdot (1 - P_c) \end{aligned} \quad (56)$$

In Formula (56)  $P_{s|x,y}$  means that system S is available given that the events  $x$  and  $y$  are true.

It has to be noted that the number of terms is  $2^n$  if  $n$  blocks are repeated. This implies that this method is tractable only for a small number of repeated blocks.

Figure 36 gives:  $P_{s|b,c} = 1$ ,  $P_{s|b,\bar{c}} = P_d$ ,  $P_{s|\bar{b},c} = P_a$ ,  $P_{s|\bar{b},\bar{c}} = 0$

Then:  $P_s = P_b \cdot P_c + P_d \cdot P_b \cdot (1 - P_c) + P_a \cdot (1 - P_b) \cdot P_c = P_a \cdot P_c + P_b \cdot P_c + P_b \cdot P_d - P_a \cdot P_b \cdot P_c - P_b \cdot P_c \cdot P_d$  (57)

#### 11.8.1.3 Karnaugh map implementation

Another way to handle the RBD presented in Figure 35 is to develop truth tables or, better, the Karnaugh map (see 11.5) which is presented in Table 8.

From this Karnaugh map the minimal success paths of the system can be found directly. They are identified by the 3 boxes drawn in Table 8.

$$s = a \bullet b + a \bullet c + b \bullet d \quad (58)$$



**Table 8 – Karnaugh map related to Figure 35**

A B \ C D	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	1	1	1
10	0	1	1	1

Therefore, the Karnaugh map is a good way to identify the success paths which have already been represented in Figure 36. This is useful from a qualitative analysis point of view but those success paths are not disjoint and this implies that the Sylvester-Poincaré formula cannot be simplified for probabilistic calculations.

**11.8.1.4 Shannon decomposition implementation**

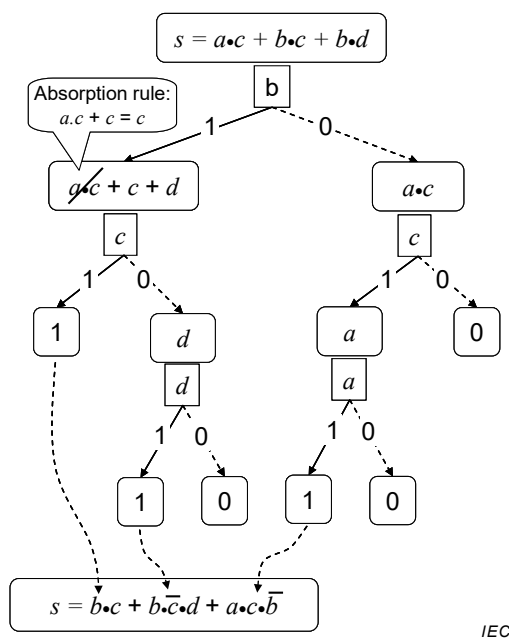
The Shannon decomposition has been done in Figure 38 and 3 disjoint success paths have been identified:

$$s = b \cdot c + b \cdot \bar{c} \cdot d + a \cdot c \cdot \bar{b} \tag{59}$$

Of course, from the Boolean algebra point of view, Formulae (59) and (58) are equivalent but Formula (59) which is made of disjoint terms leads directly to the system availability:

$$A_S(t) = A_B(t) \cdot A_C(t) + A_B(t) \cdot [1 - A_C(t)] \cdot A_D(t) + A_A(t) \cdot A_C(t) \cdot [1 - A_B(t)] \tag{60}$$

The result of the Shannon decomposition depends on the order of the variables used to develop it and therefore other equivalent expressions can be found when the order of variable changes (see B.7 where the same RBD has been analysed).



IEC

**Figure 38 – Shannon decomposition related to Figure 35**

### 11.8.2 $m$ out of $n$ models (non-identical items)

The procedure described in 9.4 is not applicable here because the blocks are not identical. As an example, consider a 2/5 system represented by the RBD in Figure 39.

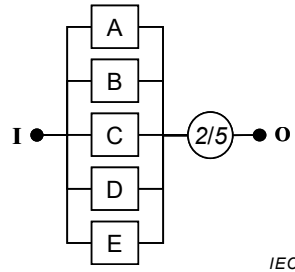


Figure 39 – 2-out-of-5 non-identical items

The availability of such a system may be evaluated by either of the techniques described in 11.3, 11.4, 11.5 or 11.6. Among them, the truth table described in 11.4 will require 32 entries from which 6 lead to the system failure:

$$(\bar{a} \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} \cdot \bar{e}), (\bar{a} \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} \cdot e), (\bar{a} \cdot \bar{b} \cdot \bar{c} \cdot d \cdot \bar{e})$$

$$(\bar{a} \cdot \bar{b} \cdot c \cdot \bar{d} \cdot \bar{e}), (\bar{a} \cdot b \cdot \bar{c} \cdot \bar{d} \cdot \bar{e}), (a \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} \cdot \bar{e})$$

Then the unavailability  $U_S$  can be derived as

$$\begin{aligned} U_S = & (1 - A_A) \cdot (1 - A_B) \cdot (1 - A_C) \cdot (1 - A_D) \cdot (1 - A_E) + (1 - A_A) \cdot (1 - A_B) \cdot (1 - A_C) \cdot (1 - A_D) \cdot A_E + \\ & (1 - A_A) \cdot (1 - A_B) \cdot (1 - A_C) \cdot A_D \cdot (1 - A_E) + (1 - A_A) \cdot (1 - A_B) \cdot A_C \cdot (1 - A_D) \cdot (1 - A_E) + \\ & (1 - A_A) \cdot A_B \cdot (1 - A_C) \cdot (1 - A_D) \cdot (1 - A_E) + A_A \cdot (1 - A_B) \cdot (1 - A_C) \cdot (1 - A_D) \cdot (1 - A_E) \end{aligned} \quad (61)$$

and so  $A_S = 1 - U_S$  can be found.

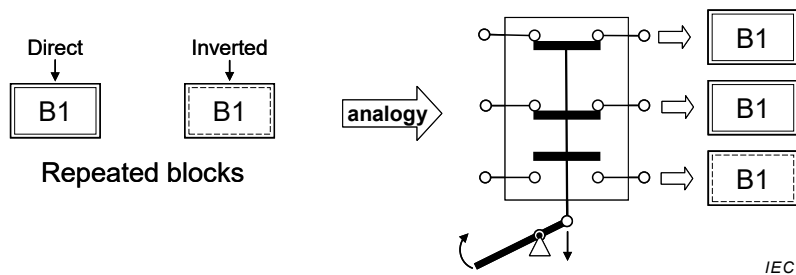
In the Formula (61),  $A_S, U_S$  and  $A_A, A_B, \dots$  can be replaced by  $A_S(t), U_S(t)$  and  $A_A(t), A_B(t), \dots$  for time dependent calculations.

## 12 Extension of reliability block diagram techniques

### 12.1 Non-coherent reliability block diagrams

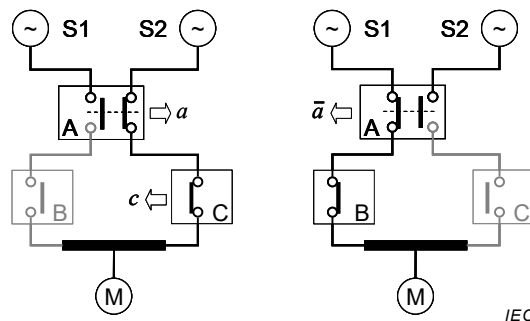
Non-coherent block diagrams are an extension from the RBDs representing monotonic logical functions to the RBDs representing non-monotonic logical functions. This may correspond, for example, to failed systems "repaired" by a further failure or to working systems failed by a further repair. This is generally unrealistic for "physical" systems but often appears when dealing with "logical" systems or with models generated automatically by model generation tools.

The main difference with an ordinary RBD is that a given block may appear in its two states (up/down). As shown in Figure 40, a new symbol has been introduced for this purpose.



**Figure 40 – Direct and inverted block**

The functioning of the "inverted" blocks can be illustrated thanks to the electrical analogy: in Figure 40, when B1 is in up state (switch closed), the inverted block B1 is in down state (switch open) and vice versa. This is also illustrated by the electrical circuit presented in Figure 41.



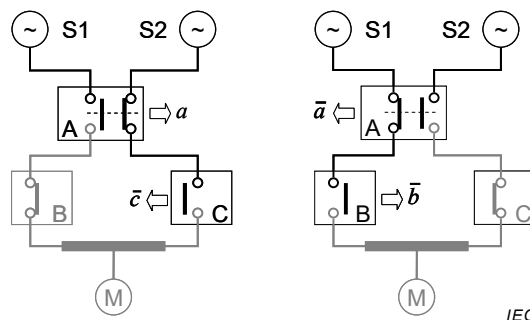
**Figure 41 – Example of electrical circuit with a commutator A**

The nominal configuration of this system is presented on the left hand side of Figure 41. With regards to the supply of motor M by S2, A is in the up state when the contact toward S2 is closed and in the down state when the same contact is open.

The motor M can be fed by the electrical source S2 or, when this is not available, by the electrical source S1. Thanks to the commutator A it cannot be fed by S1 and S2 at the same time in order to prevent short circuits between the sources.

Figure 41 highlight the two success paths allowing to feed the motor M:

- A is switched to S2 and the switch C is closed:  $a \bullet c$ ;
- A is switched to S1 and the switch B is closed:  $\bar{a} \bullet b$ .

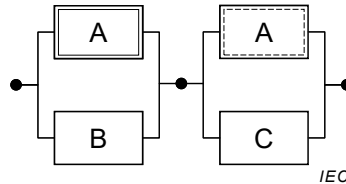


**Figure 42 – Electrical circuit: failure paths**

In the same way, Figure 42 shows the failure paths of the same electrical system:

- A is switched to S2 and the switch C is open:  $a \bullet \bar{c}$  ;
- A is switched to S1 and the switch B is open:  $\bar{a} \bullet \bar{b}$  .

A, B and C being two state components, this electrical system can be modelled by the RBD shown in Figure 43 (for the sake of simplicity, S1, S2 and M are considered to be perfect). This RBD, based on failure paths, implements repeated blocks in both direct and inverted states to model the two positions of commutator A.



**Figure 43 – Example RBD with blocks with inverted states**

The logical structure in Figure 43 is rather general and can be found in other situations. For example, the state of block A can impact the value of a physical parameter  $\theta$ :

- when block A is in up state (nominal functioning),  $\theta$  is lower than a given threshold  $\Theta$ , B is inhibited and makes C able to work;
- when block A fails then  $\theta$  becomes greater than  $\Theta$  and this inhibits the functioning of C and makes B able to work.

This RBD corresponds to the following logical equation:  $s = (a + b) \bullet (\bar{a} + c)$ .

This equation provides the three success paths:  $(a \bullet c), (\bar{a} \bullet b), (b \bullet c)$ .

This shows that, in addition to the two success paths identified above in Figure 41, a third one exists:  $b \bullet c$ . In this case the state of A does not matter.

Contrarily to the ordinary case, the success paths are not made only with blocks in up state. This produces the following side effects:

- when this system is, for example, in the up state,  $\bar{a} \bullet b \bullet \bar{c}$ , it goes to the down state  $a \bullet b \bullet \bar{c}$  when A goes to the up state. In other words the success state  $\bar{a} \bullet b \bullet \bar{c}$  is failed when A is repaired;
- when the system is, for example, in the down state,  $a \bullet b \bullet \bar{c}$ , it goes to the up state  $\bar{a} \bullet b \bullet \bar{c}$  when A goes to the down state. In other words the failed state  $a \bullet b \bullet \bar{c}$  is repaired when A fails.

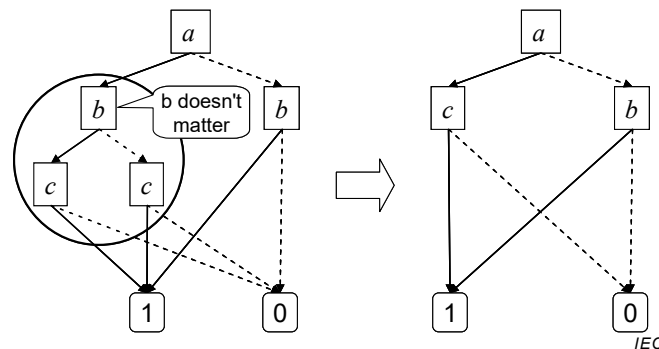
This counter intuitive behaviour is a typical property of non-coherent RBDs modelling non-monotonic logical functions. This leads to difficulties when minimal success or failure paths are needed for qualitative analysis. For example, the two failure paths,  $(\bar{a} \bullet \bar{b})$  and  $(a \bullet \bar{c})$ , are easily identified from Figure 43 but not the third one,  $(\bar{b} \bullet \bar{c})$ . This last one is not really evident and cannot be found by the classical minimal cut sets algorithms.

For non-coherent RBDs, the minimal cut sets or minimal tie sets concepts do not hold anymore. They should be superseded by the concept of prime implicants and

- the three success paths  $(a \bullet c), (\bar{a} \bullet b), (b \bullet c)$  are three prime implicants related to the success of the system,

- the tree failure paths  $(\bar{a} \bullet \bar{b}), (a \bullet \bar{c}), (\bar{b} \bullet \bar{c})$  are three prime implicants related to the failure of the system.

This implies that the popular algorithms based on the use of minimal cut sets or minimal tie sets are no longer valid for the probabilistic calculations of non-coherent RBDs.



**Figure 44 – BDD equivalent to Figure 43**

Fortunately and as shown in Figure 44, the BDD approach can be easily implemented for non-coherent RBDs. There is no difference with a BDD built for a coherent RBD and the BDD in Figure 44 can be used for the probabilistic calculation of the RBD presented in Figure 43 exactly in the same way as if it was coherent. Therefore, the use of BDDs overcomes the difficulties encountered when using minimal tie or cut sets. Nevertheless, RBD software packages are seldom able to handle prime implicants.

## 12.2 Dynamic reliability block diagrams

### 12.2.1 General

The dynamic reliability block diagram (DRBD) is an extension of common RBDs to RBDs implementing blocks interacting between themselves or with external elements. The purpose is similar to that of dynamic fault trees (see references [16] and [17]) but from the success point of view.

The RBDs developed in the previous chapters to model repaired systems (e.g. the RBD driven Markov processes described in C.4) are obviously dynamic models but the term DRBD is generally used to name RBDs fulfilling all the basic assumptions of Clause 5 except the last one concerning block independency (see 5.2 d)).

Some dynamic interactions have already been encountered in this standard when standby redundancies,  $m/n$  structures or reliability calculations of repaired systems have been dealt with.

More work has been done on dynamic fault trees than on dynamic RBDs but the problems are similar. Therefore, this standard proposes to adapt and use graphical symbols usually used for dynamic fault trees.

Some types of dynamic interactions are analysed in literature [15] but they are virtually endless. The effect of such interactions may be

- local: the states of the blocks are impacted but the logical rules of ordinary RBDs are still valid for establishing the whole system state,
- systemic: the logical rules of ordinary RBDs are no longer valid for establishing the whole system state.

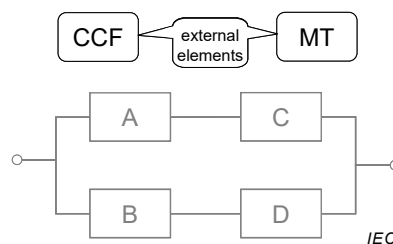
### 12.2.2 Local interactions

Those interactions which impact only the states of the blocks can be split into the following categories:

- interactions between blocks of the DRBD;
- interactions between elements external to the DRBD and blocks of the DRBD.

An event occurring on one block or on an external element (sometimes named the trigger event) interacts with the behaviour of one or several other blocks of the DRBD.

A new symbol is needed to make the difference between the external elements which do not belong to the structure of the RBD and the blocks belonging to the RBD itself. It is presented in Figure 45 where a common cause failure (CCF) and a maintenance team (MT) have been represented.



**Figure 45 – Symbol for external elements**

Examples of local interactions are the following:

- Functional dependency:
  - common cause failure: when a CCF occurs, all the related blocks fail immediately. This constitutes a strong functional dependency which can be modelled as illustrated in Figure 46;
  - loss of energy: when the energy is lost, all the related blocks stop immediately (down state). This also constitutes a strong functional dependency;
  - repair team dependency: if several blocks are repaired by the same repair team, then a failing block has to wait to be repaired if the repair team is busy with another failed block. This constitutes a functional dependency which can be modelled as illustrated in Figure 48;
  - collective repair: several blocks are repaired within the same repair operation;
  - standby redundancy: when the active block fails, this starts the standby block (see Figure 11);
  - spare parts: when an active block fails, the repair may need to use some spare parts. Therefore, the repair is possible only if one spare part is available. In addition, the spare part used to repair one block becomes unavailable to repair another block;
  - blocks in series: when one of the blocks in series fails or is under repair, the others may be stopped (e.g. because the output of the failed block is needed for later blocks in the series).
  - etc.
- Events able to occur only in a given order (an event cannot occur before another one has occurred):
  - the repair of a block cannot start before the block has failed. Such functional dependency has been already handled with ordinary RBDs for availability, reliability and frequency calculations;
  - for a given set of blocks ( $B_1, B_2, \dots, B_n$ ), the repair starts only when all of them have failed;

- the blocks become non-repairable after the whole system has failed. This is what happens when reliability calculations are performed;
- more generally, for a set of events  $(e_1, e_2, \dots, e_n)$ , this implies that  $e_2$  cannot occur as long as  $e_1$  has not occurred, that  $e_3$  cannot occur as long as  $e_2$  has not occurred, ..., that  $e_n$  cannot occur as long as  $e_{n-1}$  has not occurred. In other words,  $e_1$  is inhibited by  $\bar{e}_2$ ,  $e_2$  is inhibited by  $\bar{e}_3$ , ...,  $e_n$  is inhibited by  $\bar{e}_{n-1}$ . Therefore, the events can only occur in the sequence  $e_1, e_2, \dots, e_n$ . This may be the case when an electrical device cannot be started before the electrical power is switched on or when a cold standby device cannot be activated before the failure of the active device. This interaction is similar to sequential gates (often noted SEQ) found in dynamic fault tree analysis (see the SEQ gate in Table 4);
- etc.

### 12.2.3 Systemic dynamic interactions

Those interactions do not necessarily imply functional dependencies between the blocks which may behave independently from each other. They occur when the ordinary logical rules cannot be used.

Examples are the following:

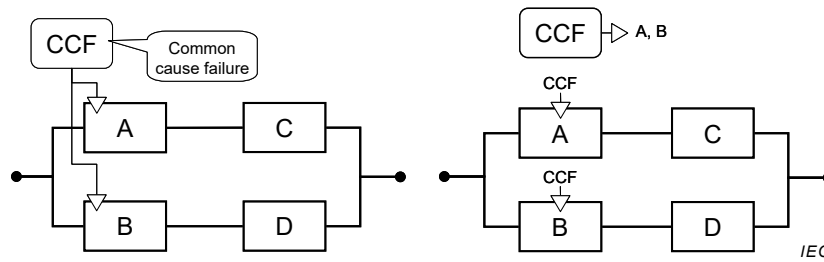
- $m/n$  majority vote: this logical configuration has already been analysed (see 7.5.1 and 9.4) and a special logical gate has been introduced to model it.
- Events which shall occur in a given order:
  - demand triggering an action performed by a given block B: if the demand occurs before B has failed, the action is performed and the system remains in up state, if the demand occurs after B has failed, the action is not performed and the system fails;
  - isolation valve protecting a system against overpressure: a hazardous event occurs only if the valve is opened before the pressure has been dropped down upstream the isolation valve;
  - more generally, for a set of events  $e_1, e_2, \dots, e_n$ , the output is produced only if the events occur in this given order, otherwise no output is produced. This interaction is similar to the "priority" AND gates (often noted PAND) found in dynamic fault tree analysis and which can also be used for DRBDs. This may be represented as a gate combining the input of several blocks.

Special gates are needed to represent the systemic dynamic dependencies as, for example, the  $m/n$  and the PAND or SEQ gates presented in Table 4 and which are popular extensions of dynamic fault trees.

The  $m/n$  gate has already been analysed and PAND and SEQ gates are analysed hereafter (see Figure 49 to Figure 52). The symbols usually implemented in dynamic fault trees have been used here but NOT gates have been inserted in inputs and outputs in order to keep the coherence with regards to the RBD logic.

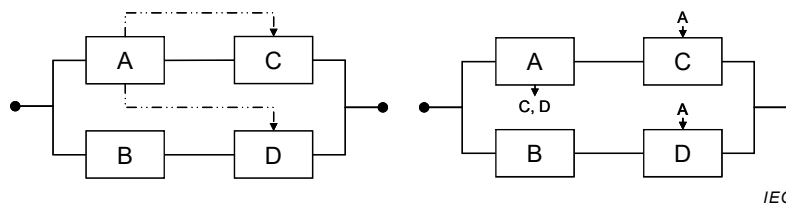
### 12.2.4 Graphical representations of dynamic interactions

As said in 12.2.2 and 12.2.3, the kinds of possible dynamic interactions are virtually endless. Therefore, even if some attempts have been made (see references [15], [16] and [17]) to propose graphical symbols for specific cases, this does not cover all the cases and only some basic graphical elements can be proposed in this standard.



**Figure 46 – Dynamic interaction between a CCF and RBDs' blocks**

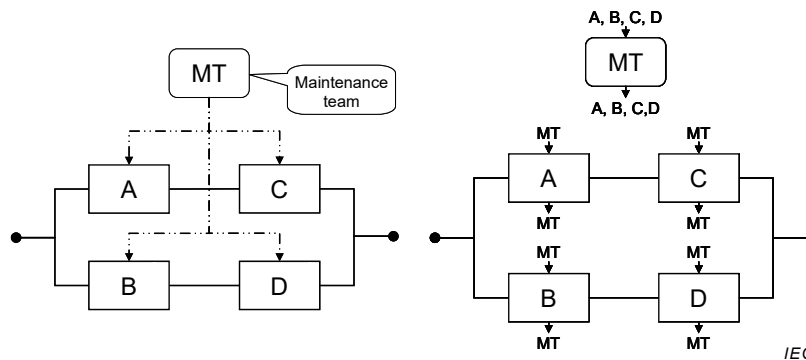
Figure 46 shows the strong interactions (i.e. strong functional dependencies) between an external element and some blocks: blocks A and B fail when the common cause failure represented by the external block CCF occurs.



**Figure 47 – Various ways to indicate dynamic interaction between blocks**

Figure 47 shows two ways to represent the interaction (i.e. functional dependencies) between blocks: the state of blocks C and D depends on the state of block A.

The same mechanisms have been implemented in Figure 48 to represent the interaction between the single repair team and the repaired blocks.



**Figure 48 – Dynamic interaction between a single repair team and RBDs' blocks**

These simple graphical representations aim only at indicating that there is some dynamic interaction between the blocks and the external elements. The dotted lines on the left hand side of Figure 47 and Figure 48 can be used when only few interactions have to be represented in an RBD. When there are many interactions to be represented, the proposal on the right hand side of Figure 47, Figure 46 and Figure 48 is clearer. The very nature of the interactions themselves should be specified elsewhere. The main use of these representations is to support the graphical presentation of the RBD and to ensure that the external elements are well identified.

Figure 49 shows how a PAND gate can be used within a DRBD: the output O goes to the down state only if  $I_1$  goes to the down state before  $I_2$  goes to the down state.



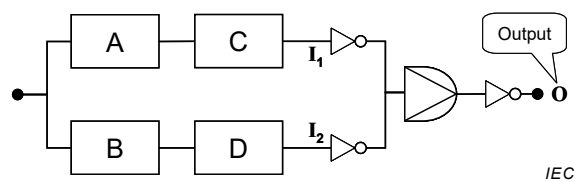


Figure 49 – Implementation of a PAND gate

The functioning of the PAND gate is illustrated in Figure 50. The PAND gate is equivalent to the 5 states of the finite-state automaton drawn on the left hand side of the figure.

- State 1:  $I_1$  and  $I_2$  are in up state. Then the output  $O$  is in the up state.
- State 2:  $I_2$  has gone to the down state first and  $I_1$  is still in the up state. Then the output  $O$  is in the up state.
- State 3:  $I_1$  has gone to the down state first and  $I_2$  is still in the up state. Then the output  $O$  is in the up state.
- State 4:  $I_1$  and  $I_2$  have gone to the down state but  $I_2$  has gone first. Then the output  $O$  is in the up state.
- State 5:  $I_1$  and  $I_2$  have gone to the down state but  $I_1$  has gone first. Then the output  $O$  has gone to the down state.

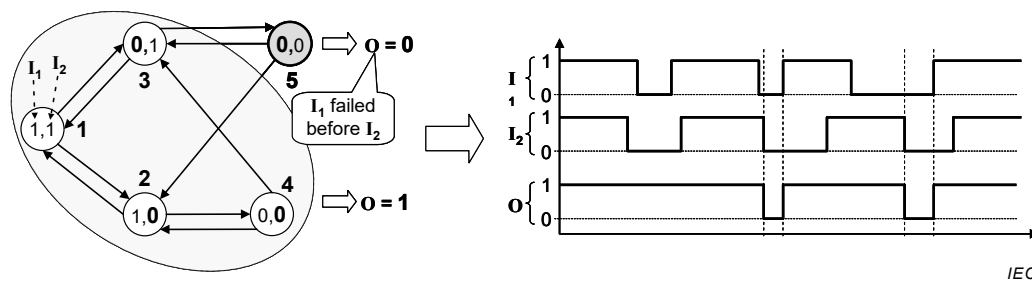


Figure 50 – Equivalent finite-state automaton and example of chronogram for a PAND gate

Then, when the input  $I_1$  and  $I_2$  varies between 1 and 0, the output of the PAND gate (Figure 49) changes according to the rules presented by this finite-state automaton. This gives, for example, the chronogram presented on the right hand side of Figure 50. A Petri net modelling the same finite-state automaton is analysed in Annex E and Figure E.6.

Figure 51 shows how a SEQ gate can be used within a DRBD: as for the PAND gate, the output  $O$  goes to the down state only if  $I_1$  goes to the down state before  $I_2$  goes to the up state. The difference is that  $I_2$  cannot go to the down state before  $I_1$  has gone to the down state first. Therefore, the failure of B and D are inhibited as long as  $I_1$  is in up state and this is indicated thanks to the dynamic interactions drawn in dotted lines.

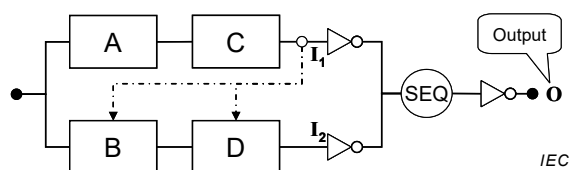
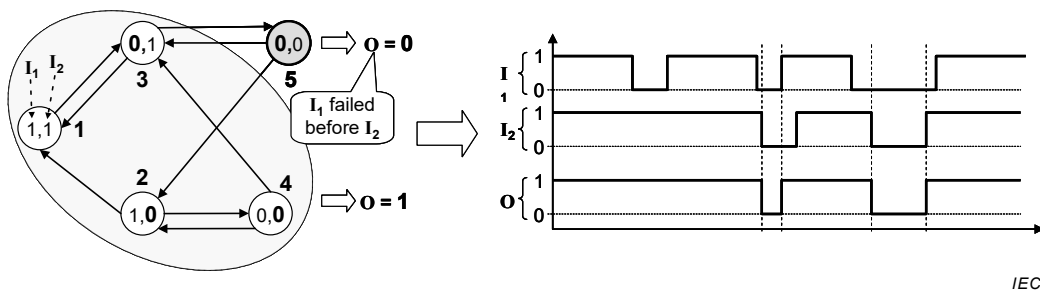


Figure 51 – Implementation of a SEQ gate

The functioning is illustrated in Figure 52. The SEQ gate is equivalent to the 5 states finite-state automaton drawn on the left hand side of the figure. The states are the same as for the

PAND gate except that there is no transition from state 1 to state 2 in order to force the order of the failures:  $I_1$  first, then  $I_2$ .



**Figure 52 – Equivalent finite-state automaton and example of chronogram for a SEQ gate**

As shown in the chronogram,  $I_2$  cannot fail before  $I_1$  has previously failed.

A Petri net modelling the same finite-state automaton is analysed in Annex E, Figure E.6 and Figure E.7.

### 12.2.5 Probabilistic calculations

Making the probabilistic calculation by using the Markovian approach is proposed in literature (see references [2], [29] and [30]). Nevertheless, building a Markov process for a whole DRBD is quickly limited by the combinatorial explosion of the number of states. Therefore, this approach should be restricted to small independent parts of the DRBD as this has been done for the RBD driven Markov processes described in Clause C.4.

Another approach which is proposed in literature is to make the link between DRBDs and finite state automata (state-events machine or Petri net). This is more effective than the markovian approach but the analytical calculations are no longer possible and Monte Carlo simulation has to be implemented.

The RBD driven Petri nets described in Annex E are an effective way to mix the RBD and PN approaches in order to deal with dynamic RBDs problems and calculations.

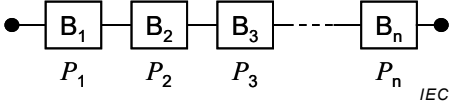
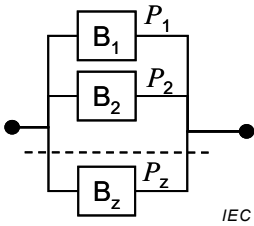
## Annex A (informative)

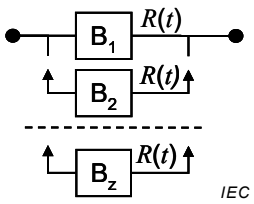
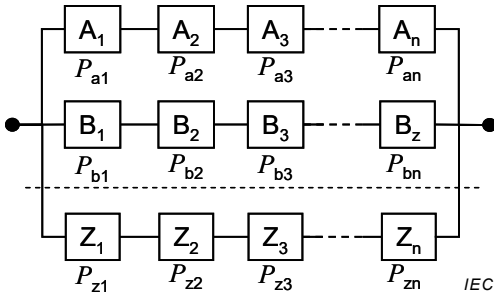
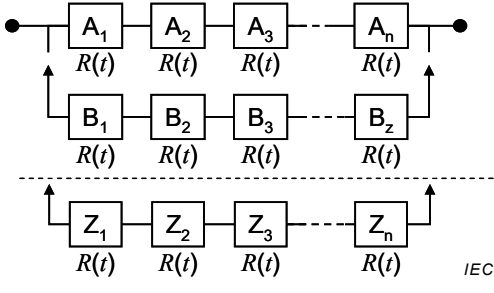
### Summary of formulae

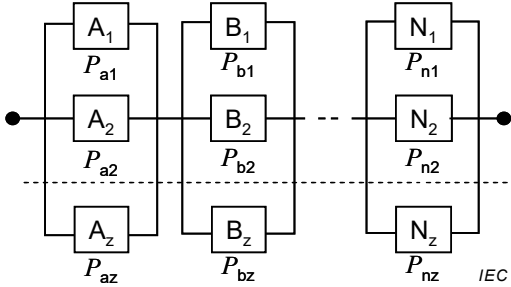
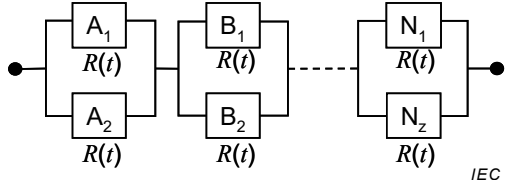
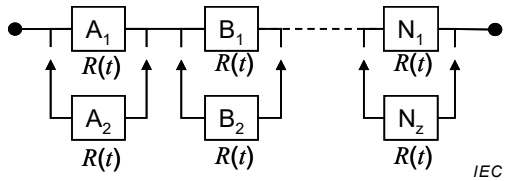
Warning: The formulae presented in Table A.1 are intended to be used by users aware of the underlying hypothesis and mathematics and of the limitations when approximations are implemented.

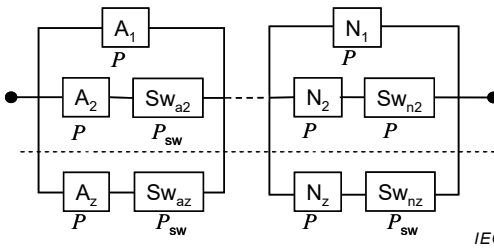
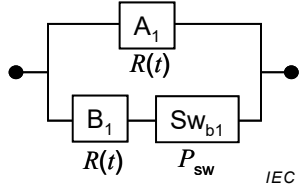
NOTE In Table A.1, frequent use is made of the terms “active” and “standby”. The former is used to indicate that the blocks concerned (each of which can consist of a component, sub-system, system, etc.) are energized (powered-up) and hence are liable to failure. The latter on the other hand is used to indicate that the block or blocks concerned are de-energized (powered-down) and not liable to failure.

**Table A.1 – Example of equations for calculating the probability of success of basic configurations**

Basic configuration	Equation for system $P_S, R_S(t), A_S(t)$
<p><b>1 Series structures</b></p> 	<p><b>A General case</b></p> <p>Constant probabilities:</p> $P_S = P_1 \cdot P_2 \dots P_n$ <p>Time dependent probabilities:</p> $R_S(t) = R_1(t) \cdot R_2(t) \dots R_n(t)$ $A_S(t) = A_1(t) \cdot A_2(t) \dots A_n(t)$
	<p><b>B</b> With <math>P_1 = P_2 = \dots P_n = P</math> =&gt; <math>P_S = P^n</math></p> <p><b>C</b> With <math>R_1(t) = R_2(t) = \dots R_n(t) = R(t)</math> =&gt; <math>R_S(t) = R(t)^n</math></p> <p><b>D</b> With <math>A_1(t) = A_2(t) = \dots A_n(t) = A(t)</math> =&gt; <math>A_S(t) = A(t)^n</math></p>
<p><b>2 Parallel structures</b></p> <p>Active</p> 	<p><b>A Active general case</b></p> <p>Constant probabilities:</p> $P_S = 1 - (1 - P_1) \cdot (1 - P_2) \dots (1 - P_z)$ <p>Time dependent probabilities:</p> <p><math>R_S(t)</math>: no simple general formula (see NOTE 1)</p> $A_S(t) = 1 - [1 - A_1(t)] \cdot [1 - A_2(t)] \dots [1 - A_z(t)]$
	<p><b>B</b> With <math>P_1 = P_2 \dots = P</math> =&gt; <math>P_S = 1 - (1 - P)^z</math></p> <p><b>C</b> With <math>A_1(t) = A_2(t) = \dots A_z(t) = A(t)</math> =&gt; <math>A_S(t) = [1 - A(t)]^z</math></p>

Basic configuration	Equation for system $P_S, R_S(t), A_S(t)$
<p><b>Standby</b></p> 	<p><b>D Standby with <math>R(t) = A(t) = e^{-\lambda \cdot t}</math> (i.e., non-repaired items)</b></p> $R_S(t) = A_S(t) = e^{-\lambda \cdot t} + \lambda \cdot t \cdot e^{-\lambda \cdot t} + \dots + \frac{(\lambda \cdot t)^{z-1} e^{-\lambda \cdot t}}{(z-1)!}$
<p><b>3 Series-parallel structures (redundant systems)</b></p> <p><b>Active</b></p> 	<p><b>A Active general case</b></p> <p>Constant probabilities</p> $P_S = 1 - \prod_{i=1}^z (1 - P_{i1} \cdot P_{i2} \cdot \dots \cdot P_{in}) = 1 - \prod_{i=1}^z [1 - \prod_{j=1}^n P_{ij}]$ <p>Time dependent probabilities:</p> <p><math>R_S(t)</math>: no simple general formula (see NOTE 1)</p> $A_S(t) = 1 - \prod_{i=1}^z [1 - \prod_{j=1}^n A_{ij}(t)]$ <p><b>B Active with</b></p> $P_{i1} = P_{i2} = \dots = P_i \quad \forall i \quad \Rightarrow P_S = 1 - \prod_{i=1}^z (1 - P_i^n)$ <p><b>C Active with</b></p> $A_{i1}(t) = A_{i2}(t) = \dots = A_i(t) \quad \forall i \quad \Rightarrow A_S(t) = 1 - \prod_{i=1}^z [1 - A_i(t)^n]$ <p><b>D Active with</b></p> $P_{ij} = P \quad \forall i, j \quad \Rightarrow P_S = 1 - \prod_{i=1}^z (1 - P^n)$ <p><b>E Active with</b></p> $A_{ij}(t) = A(t) \quad \forall i, j \quad \Rightarrow A_S(t) = 1 - [1 - A(t)^n]^z$
<p><b>Standby</b></p> 	<p><b>F Standby with <math>R(t) = A(t) = e^{-\lambda t}</math> (i.e., non-repaired items)</b></p> $R_S(t) = A_S(t) = e^{-n\lambda t} + n\lambda t \cdot e^{-n\lambda t} + \dots + \frac{(n\lambda t)^{z-1} e^{-n\lambda t}}{(z-1)!}$

Basic configuration	Equation for system $P_S, R_S(t), A_S(t)$
<p><b>4 Parallel-series structures</b> (redundant elements)</p>	
<p><b>Active</b></p>  	<p><b>A Active general case</b> Constant probabilities</p> $P_S = \prod_{i=a}^n \left\{ 1 - \prod_{j=1}^z (1 - P_{ij}) \right\}$ <p>Time dependent probabilities: <math>R_S(t)</math>: no simple general formula (see NOTE 1)</p> $A_S(t) = \prod_{i=a}^n \left\{ 1 - \prod_{j=1}^z [1 - A_{ij}(t)] \right\}$ <p><b>B Active with</b> <math>P_{i1} = P_{i2} = \dots = P_{iz} = P_i \quad \forall i \quad \Rightarrow P_S = \prod_{i=a}^n [1 - (1 - P_i)^z]</math></p> <p><b>E Active with</b> <math>A_{i1}(t) = A_{i2}(t) = \dots = A_{iz}(t) = A_i(t) \quad \forall i</math> <math>\Rightarrow A_S(t) = \prod_{i=a}^n \{1 - [1 - A_i(t)]^z\}</math></p> <p><b>F Active with</b> <math>P_{ij} = P \quad \forall i, j \quad \Rightarrow P_S = [1 - (1 - P)^z]^n</math></p> <p><b>G Active with</b> <math>A_{ij}(t) = A(t) \quad \forall i, j \quad \Rightarrow A_S(t) = \{1 - [1 - A(t)]^z\}^n</math></p> <p><b>H Assuming <math>R(t) = A(t) = e^{-\lambda \cdot t}</math></b> <math>R_S(t) = A_S(t) = (2 \cdot e^{-\lambda \cdot t} - e^{-2 \cdot \lambda \cdot t})^n</math></p>
<p><b>Standby</b></p> 	<p><b>D Standby with <math>R(t) = A(t) = e^{-\lambda \cdot t}</math></b> <math>R_S(t) = A_S(t) = (e^{-\lambda \cdot t} + \lambda \cdot t \cdot e^{-\lambda \cdot t})^n</math></p>

Basic configuration	Equation for system $P_S, R_S(t), A_S(t)$
<p><b>5 Parallel-series structures</b> (redundant elements)</p>  	<p><b>A</b> Active with <math>P_{ij} = P \quad \forall i, j</math> except <math>P_{sw}</math></p> $\Rightarrow P_S = [1 - (1 - P) \cdot (1 - P \cdot P_{sw})^{z-1}]^n$ <p><b>B</b> Active with <math>A_{ij}(t) = A(t) \quad \forall i, j</math> except <math>A_{sw}(t)</math></p> $\Rightarrow A_S(t) = \left\{ 1 - [1 - A_S(t)] \cdot [1 - A_S(t) \cdot A_{sw}(t)]^{z-1} \right\}^n$ <p><b>C</b> Active with <math>z = 2, n = 1</math></p> <p>and <math>R_{ij}(t) = A_{ij}(t) = e^{-\lambda t} \quad \forall i, j</math> except <math>P_{sw}</math></p> $\Rightarrow R_S(t) = A_S(t) = e^{-\lambda t} + P_{sw} e^{-\lambda t} - P_{sw} e^{-2\lambda t}$
<p>NOTE 1 In case of non-repaired blocks <math>R_S(t) = A_S(t)</math>.</p> <p>NOTE 2 For non-repaired blocks with constant failure rates, <math>P</math> can be replaced by <math>R(t) = A(t) = e^{-\lambda t}</math>.</p> <p>NOTE 3 Formulae for standby systems are based on the assumption that the reliability of switching and sensing mechanisms is 100 % (<math>P_{sw} = 1</math>).</p>	

## Annex B (informative)

### Boolean algebra methods

#### B.1 Introductory remarks

Apart from the use of Boolean truth tables (see 11.4) and binary decision diagrams (see 11.5), the analysis of RBDs as described so far makes use mainly of conventional algebraic mathematical formulae. However, Boolean algebra in general can also be used for such analyses, and in many instances is much more efficacious and straightforward. In particular, the use of Boolean algebra may well be the most straightforward approach whenever

- a) RBDs contain repeated blocks (see Figure 37),
- b) RBDs contain directional arrows (see Figure 10 and Figure 35),
- c) the system is particularly complicated,
- d) it is easier to construct a Boolean expression for system success (or failure) than it is to construct an RBD,
- e) the system comprises a number of blocks too large to be tractable by simple formulae.

Item d) of the above list is worthy of note. For many systems and networks the listing of equipment success (or failure) combinations in Boolean terms is often a more straightforward task than the construction of the corresponding RBD. By employing at the outset the Boolean approach to analyse the system, the risk of making errors in the course of constructing the RBD is entirely avoided.

Item e) of the above list may be related to RBDs modelling industrial systems with a dozen of components and leading to the combinatorial explosion of the terms to be taken into account in the formulae. This is particularly crucial when numerous repeated blocks also have to be managed.

#### B.2 Notation

The conventional symbols  $\cup$  and  $\cap$  denoting the logical “OR” and “AND” play for the Boolean algebra the same role as the addition (+) and of the multiplication ( $\cdot$ ) for ordinary algebra. This is why, in what follows, it has been found more convenient, to use a “+” symbol to denote logical “OR” and a full stop “•” to denote logical “AND”<sup>2</sup>. As usual a bar over a Boolean variable will denote the inverse or complement of the variable concerned: e.g.  $\bar{a}$  is interpreted as “not  $a$ ”. For example  $a \cdot b \cdot \bar{c} \cdot e + f \cdot g$  is to be interpreted “ $a$  AND  $b$  AND NOT  $c$  AND  $e$  OR  $f$  AND  $g$ ”. The context in which the symbols are used should make the meaning clear.

---

<sup>2</sup> The advantage of such a notation becomes apparent in Annex B where expressions of the type  $S = a \cdot b + \bar{a} \cdot e \cdot b + \bar{a} \cdot e \cdot d + a \cdot \bar{b} \cdot e \cdot d + \bar{a} \cdot c \cdot d + a \cdot \bar{b} \cdot c \cdot d$  are frequently found. Taking this latter expression as an example and writing it using set theory symbols, one obtains:  $S = a \cap b \cup \bar{a} \cap e \cap b \cup \bar{a} \cap e \cap d \cup a \cap \bar{b} \cap e \cap d \cup \bar{a} \cap c \cap d \cup a \cap \bar{b} \cap c \cap d$  which for many readers may be quite difficult to interpret or evaluate.

### B.3 Tie sets (success paths) and cut sets (failure paths) analysis

#### B.3.1 Notion of cut and tie sets

As said in 8.1, an RBD can be considered as an electrical circuit (see Figure 14) and this analogy is useful to identify:

- the tie sets which correspond to a closed electrical circuit and represent the combinations of the blocks in up states leading to the system being in the up state. The tie sets are also the "success" paths of the RBD;
- the cut sets which correspond to a cut electrical circuit and represent the combinations of the blocks in down states leading to the system being in the down state. The cut sets are also the "failure" paths of the RBD.

Using this analogy allows to transform the RBD presented in Figure 10 into the electrical circuit presented in Figure 15. From this representation it is easy to identify various tie sets of this RBD and Figure B.1 and Figure B.2 show various examples of combinations of closed switches corresponding to the system up state.

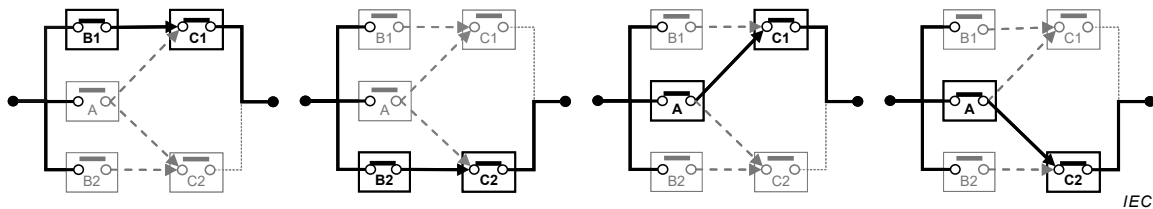


Figure B.1 – Examples of minimal tie sets (success paths)

In Figure B.1 any opening (i.e. any failure) of the closed switches will cut the circuit and lead to the system down state. All those closed switches (i.e. blocks in up states) are necessary and sufficient to have the system in up state. Those combinations are minimal and are named minimal tie sets.

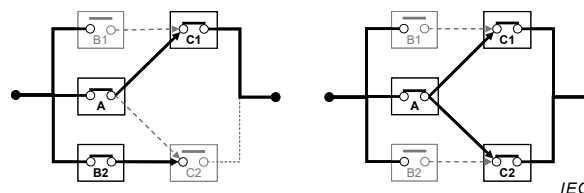


Figure B.2 – Examples of non-minimal tie sets (non minimal success paths)

In Figure B.2 some opening of the closed switches (e.g. B2 on the left or C1 on the right) will not change the system up state. All the closed switches (i.e. blocks in up states) are not necessary to have the system in up state. Those combinations are not minimal and are named non-minimal tie sets (or ordinary tie sets).

From the same Figure 15 it is also easy to identify various cut sets of this RBD and Figure B.3 and Figure B.4 show various examples of combinations of open switches corresponding to the system down state.



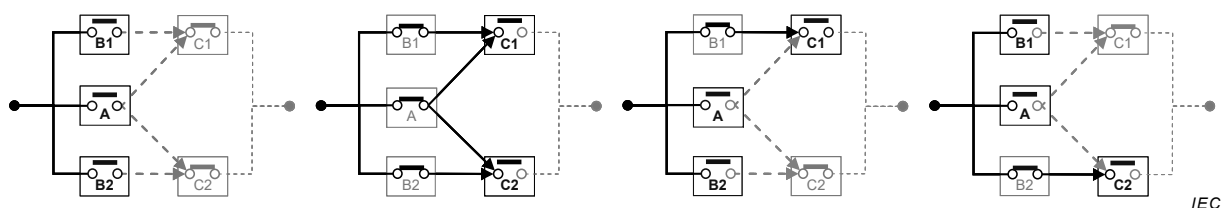


Figure B.3 – Examples of minimal cut sets

In Figure B.3 any closing (i.e. any repair) of the open switches will close the circuit and lead to the system up state. All the open switches (i.e. blocks in down states) are necessary and sufficient to have the system in down state. Those combinations are minimal and are named minimal cut sets.

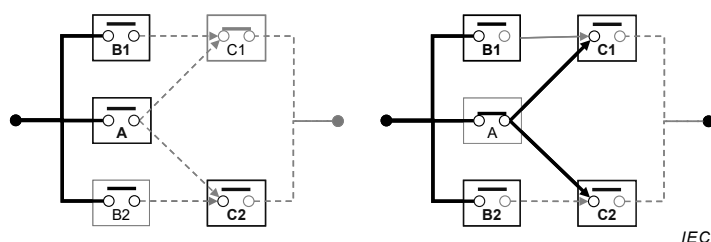


Figure B.4 – Examples of non-minimal cut sets

In Figure B.4 some closing of the open switches (e.g. C2 on the left or B2 on the right) will not change the system down state. All the open switches (i.e. blocks in down states) are not necessary to have the system in down state. Those combinations are not minimal and are named non-minimal cut sets (or ordinary cut sets).

### B.3.2 Series-parallel representation using minimal tie and cut sets

Applying the Boolean algebra properties leads to represent the system up state,  $s$ , as the union of the minimal tie sets ( $\Pi_i$ ) of the RBD and the system down state,  $\bar{s}$ , as the union of the minimal cut sets ( $C_k$ ) of the RBD.

This can be applied to the previous example which has four minimal tie sets,  $(b_1 \bullet c_1)$ ,  $(a \bullet c_1)$ ,  $(a \bullet c_2)$ ,  $(b_2 \bullet c_2)$ . This leads to:

$$s = \bigcup_i \Pi_i = b_1 \bullet c_1 + a \bullet c_1 + a \bullet c_2 + b_2 \bullet c_2 \quad (\text{B.1})$$

The same example has four minimal cut sets,  $(\bar{b}_1 \bullet \bar{a} \bullet \bar{b}_2)$ ,  $(\bar{c}_1 \bullet \bar{c}_2)$ ,  $(\bar{b}_1 \bullet \bar{a} \bullet \bar{c}_2)$ ,  $(\bar{b}_2 \bullet \bar{a} \bullet \bar{c}_1)$  and this leads to:

$$\bar{s} = \bigcup_j C_j = \bar{b}_1 \bullet \bar{a} \bullet \bar{b}_2 + \bar{c}_1 \bullet \bar{c}_2 + \bar{b}_1 \bullet \bar{a} \bullet \bar{c}_2 + \bar{b}_2 \bullet \bar{a} \bullet \bar{c}_1 \quad (\text{B.2})$$

These formulae provide "dual" representations of the same system. Formula (B.1) is focused on the system success when Formula (B.2) is focused on system failure.

Formula (B.2) is equivalent to  $s = \bar{\bar{s}} = \overline{\bigcup_j C_j} = \bar{b}_1 \bullet \bar{a} \bullet \bar{b}_2 + \bar{c}_1 \bullet \bar{c}_2 + \bar{b}_1 \bullet \bar{a} \bullet \bar{c}_2 + \bar{b}_2 \bullet \bar{a} \bullet \bar{c}_1$ .

The transformation of Formula (B.2) involves the use of the De Morgan laws:

$$\overline{a \bullet b} = \overline{a} + \overline{b}$$

$$\overline{a + b} = \overline{a} \bullet \overline{b}$$

This leads to:

$$s = \overline{b_1 \bullet \overline{a} \bullet b_2 \bullet \overline{c_1} \bullet \overline{c_2} \bullet \overline{b_1} \bullet \overline{a} \bullet c_2 \bullet \overline{b_2} \bullet \overline{a} \bullet \overline{c_1}} \text{ which gives:}$$

$$s = (b_1 + a + b_2) \bullet (c_1 + c_2) \bullet (b_1 + a + c_2) \bullet (b_2 + a + c_1) \quad (\text{B.3})$$

Finally two equivalent representations of the logical formula representing the RBD in the up state are obtained. Formula (B.1) provides a representation with the success paths and Formula (B.3) (see Figure 18) a representation of the RBD with the minimal cut sets (see Figure 19).

### B.3.3 Identification of minimal cuts and tie sets

The minimal cut sets and minimal tie sets can be obtained by expanding the logical formulae corresponding to the RBD.

This can be done with a simple RBD as shown hereafter with the RBD drawn in Figure B.5.

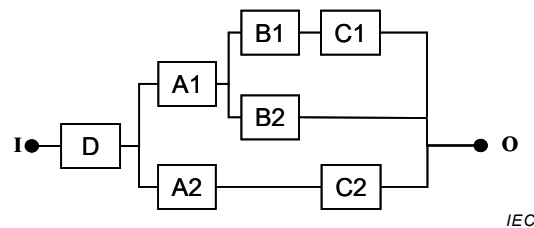


Figure B.5 – Example of RBD with tie and cut sets of various order

The logical structure of this RBD provides the following logical formula:

$$s = d \bullet \{ (a_2 \bullet c_2) + [a_1 \bullet (b_1 \bullet c_1) + a_1 \bullet b_2] \} \quad (\text{B.4})$$

Then expanding Formula (B.4) leads to:

$$s = d \bullet \{ (a_2 \bullet c_2) + [a_1 \bullet b_1 \bullet c_1 + a_1 \bullet b_2] \} \text{ and } s = d \bullet a_2 \bullet c_2 + d \bullet a_1 \bullet b_1 \bullet c_1 + d \bullet a_1 \bullet b_2.$$

Therefore this RBD has three minimal success paths:  $(d \bullet a_2 \bullet c_2)$ ,  $(d \bullet a_1 \bullet b_2)$ ,  $(d \bullet a_1 \bullet b_1 \bullet c_1)$ .

The minimal cut sets can be obtained by complementing Formula (B.4) and using the De Morgan laws:

$$\overline{s} = \overline{d \bullet \{ (a_2 \bullet c_2) + [a_1 \bullet (b_1 \bullet c_1) + a_1 \bullet b_2] \}}$$

$$\overline{s} = \overline{d} + \overline{(a_2 \bullet c_2) \bullet [a_1 \bullet (b_1 \bullet c_1) + a_1 \bullet b_2]} = \overline{d} + (\overline{a_2} + \overline{c_2}) \bullet (\overline{a_1} + \overline{b_1} + \overline{c_1}) \bullet (\overline{a_1} + \overline{b_2})$$

$$\overline{s} = \overline{d} + \overline{a_1} \bullet \overline{a_2} + \overline{a_1} \bullet \overline{c_2} + \overline{a_2} \bullet \overline{b_1} \bullet \overline{b_2} + \overline{a_2} \bullet \overline{c_1} \bullet \overline{b_2} + \overline{c_2} \bullet \overline{c_1} \bullet \overline{b_2} + \overline{c_2} \bullet \overline{b_1} \bullet \overline{b_2}$$

And finally, seven minimal cut sets are found:  $\bar{d}$ ,  $(\bar{a}_1 \bullet \bar{a}_2)$ ,  $(\bar{a}_1 \bullet \bar{c}_2)$ ,  $(\bar{a}_2 \bullet \bar{b}_1 \bullet \bar{b}_2)$ ,  $(\bar{a}_2 \bullet \bar{c}_1 \bullet \bar{b}_2)$ ,  $(\bar{c}_2 \bullet \bar{c}_1 \bullet \bar{b}_2)$ ,  $(\bar{c}_2 \bullet \bar{b}_1 \bullet \bar{b}_2)$ .

Minimal tie and cut sets represent the same information but, from a qualitative analysis point of view, the minimal cut sets are more relevant because the shortest minimal cut sets are likely to be more probable than the other minimal cut sets.

Therefore, the cut sets of the above RBD in Figure B.5 can be sorted by order (see 3.18, Note 2 to entry):

- order one,  $\bar{d}$ ;
- order two,  $(\bar{a}_1 \bullet \bar{a}_2)$ ,  $(\bar{a}_1 \bullet \bar{c}_2)$ ;
- order three,  $(\bar{a}_2 \bullet \bar{b}_1 \bullet \bar{b}_2)$ ,  $(\bar{a}_2 \bullet \bar{c}_1 \bullet \bar{b}_2)$ ,  $(\bar{c}_2 \bullet \bar{c}_1 \bullet \bar{b}_2)$ ,  $(\bar{c}_2 \bullet \bar{b}_1 \bullet \bar{b}_2)$ .

From a qualitative point of view, the weak point of this system is certainly the minimal cut set of order one,  $\bar{d}$ .

Except in simple cases, the above calculations are not really tractable by hand but powerful algorithms are available and implemented into RBD software packages. Minimal tie and cut sets may be found by using, for example, the binary decision diagrams explained in the probabilistic calculation part of this standard.

## B.4 Principles of calculations

### B.4.1 Series structures

Consider a system made of  $n$  blocks ( $B_i$ ) in series similar to that depicted in Figure 2. For that system, it can be seen that the system as a whole is in up state provided all of the blocks  $B_i$  are in up states. In other words, the Boolean expression for system success is given by

$$s = \bigcap_{i=1,n} b_i \equiv b_1 \bullet b_2 \bullet b_3 \bullet \dots \bullet b_n \quad (\text{B.5})$$

where  $b_i$  is a Boolean variable corresponding to the up state of blocks  $B_i$ .

If the blocks are independent then the probability of the system to be in up state is:

$$P_s = P_{b_1} \cdot P_{b_2} \cdot P_{b_3} \cdot \dots \cdot P_{b_n} = \prod_i P_{b_i} \quad (\text{B.6})$$

Therefore, there are no particular calculation problems for calculating  $P_s$  in case of series structures.

Nevertheless, if the above series structure (B.5) belongs to a larger RBD, this calculation can be done only if no block of this series structure is repeated elsewhere in the larger RBD. Otherwise the techniques described in Clauses B.5 or B.6 should be applied

### B.4.2 Parallel structures

Consider a two unit active redundant system such as that depicted in Figure 21. For that system, it can be seen that the system as a whole is in up state provided A or B (or both) are in up states. In other words, the Boolean expression for system success is given by

$$s = a \cup b \equiv a + b \quad (\text{B.7})$$

where  $a$  and  $b$  are Boolean variables corresponding to the up state of blocks A and B respectively.

For a given time  $t$  it is tempting to substitute  $P_a$  and  $P_b$  for  $a$  and  $b$  respectively and rewrite Formula (B.7) in the form:

$$P_s = P_a + P_b \quad (\text{B.8})$$

Formula (B.8) is expected to provide the probability  $P_s$  that the system is in the up state but, unfortunately, is incorrect owing to the fact it is obtained from a Boolean expression in which the variables overlap (i.e.  $a \cap b \equiv a \bullet b \neq \Phi$ ). It does not even provide, in the general case, an acceptable approximation of  $P_s$ . For example  $P_s = 1,2$  is obtained with  $P_a$  and  $P_b$  equal to 0,6. This is obviously incorrect.

Therefore Formula (B.8) shall be completed to:

$$P_s = P_a + P_b - P_a \cdot P_b \quad (\text{B.9})$$

Contrarily to Formula (B.8), Formula (B.9) provides the exact result in any case. With the previous figures it leads to:  $P_s = 0,6 + 0,6 - 0,36 = 0,84$ .

Considering a structure made of  $n$  blocks ( $B_i$ ) in parallel leads to:

$$s = \bigcup_i^n b_i = b_1 + b_2 + b_3 + \dots + b_n$$

The extension of Formula (B.9) is known as the Sylvester-Poincaré formula:

$$P_s = P\left(\bigcup_{i \leq n} b_i\right) = \sum_{i \leq n} P_{b_i} - \sum_{i < j \leq n} P_{b_i} \cdot P_{b_j} + \sum_{i < j < k \leq n} P_{b_i} \cdot P_{b_j} \cdot P_{b_k} - \text{etc.} \quad (\text{B.10})$$

This is an alternate sum of decreasing terms which converge toward the result  $P_s$ . The number of terms drastically increases when  $n$  increases and the convergence is very slow when the probabilities are high. This is unfortunately the case here because the probabilities,  $P_{b_i}$ , for the blocks to be in up state are normally close to 1. Therefore, Formula (B.10) is not manageable to evaluate  $P_s$  because too many terms need to be taken into consideration.

Fortunately, several alternatives can be considered. The first one is to evaluate the probability for the system to be in the down state rather than in the up state.

For example, the "down state" of the small system (B.7) analysed above is given by  $\bar{s} = \overline{a + b}$  which, by applying the De Morgan laws, leads to the equivalent dual form of the Boolean expression  $\bar{s} = \bar{a} \bullet \bar{b}$ .

If  $a$  and  $b$  are independent from each other, this is the same for  $\bar{a}$  and  $\bar{b}$ . Then the probability for the system to be in the down state is given by:  $P_{\bar{s}} = P_{\bar{a}} \cdot P_{\bar{b}}$ .

And finally:  $P_s = 1 - P_{\bar{s}} = 1 - (1 - P_a) \cdot (1 - P_b)$ .

This can easily be extended to  $n$  blocks in parallel and  $s = \bigcup_{i=1,n} b_i \Leftrightarrow \bar{s} = \bigcap_{i=1,n} \bar{b}_i$  is obtained by applying the De Morgan laws.

$$\text{And finally: } P_s = P\left(\bigcup_i b_i\right) = 1 - P\left(\bigcap_{i=1,n} \bar{b}_i\right) = 1 - \prod_{i=1,n} P_{\bar{b}_i} = 1 - \prod_{i=1,n} (1 - P_{b_i}) \quad (\text{B.11})$$

Formula (B.11) which involves only simple products is easier to use for parallel structures than the Sylvester-Poincaré formula analysed just above.

### B.4.3 Mix of series and parallel structures

Formulae (B.6) and (B.11) can be combined and this can be done by hand in simple cases.

Thus, if a system exists as depicted by Figure 4 but with only three items in each branch, the probability of success of the system is:

$$P_s = P_{a1} \cdot P_{b1} \cdot P_{c1} + P_{a2} \cdot P_{b2} \cdot P_{c2} - P_{a1} \cdot P_{b1} \cdot P_{c1} \cdot P_{a2} \cdot P_{b2} \cdot P_{c2} \quad (\text{B.12})$$

Similarly, for Figure 5, the following applies:

$$P_s = (P_{a1} + P_{a2} - P_{a1} \cdot P_{a2}) \cdot (P_{b1} + P_{b2} - P_{b1} \cdot P_{b2}) \cdot (P_{c1} + P_{c2} - P_{c1} \cdot P_{c2}) \quad (\text{B.13})$$

For Figure 6 and Figure 7, the probability of success of the system equations are obtained simply by multiplying Formulae (B.12) and (B.13) by  $P_d$ .

Nevertheless and except in simple cases, the above calculations (B.12) or (B.13) are not easily tractable by hand. Fortunately, powerful algorithms are available and implemented into RBD software packages. They are based on the techniques described in Clauses B.5, B.6 or B.7.

### B.4.4 $m$ out of $n$ architectures (identical items)

Among the simple cases, the formulae related to the probability of success of systems corresponding to Figure 8 (2/3 logics) and Figure 9 (3/4 logics) are a little more complicated than those developed in B.4.3.

Looking at the 2/3 system presented in Figure 8, Formula (B.9) leads to:

$$\begin{aligned} P_s &= P(x_1 \bullet x_2 + x_1 \bullet x_3 + x_2 \bullet x_3) = P(x_1 \bullet x_2 + x_1 \bullet x_3) + P(x_2 \bullet x_3) - P(x_1 \bullet x_2 \bullet x_3) \\ &= P(x_1 \bullet x_2) + P(x_1 \bullet x_3) - P(x_1 \bullet x_2 \bullet x_3) + P(x_2 \bullet x_3) - P(x_1 \bullet x_2 \bullet x_3) \\ &= P(x_1 \bullet x_2) + P(x_1 \bullet x_3) + P(x_2 \bullet x_3) - 2P(x_1 \bullet x_2 \bullet x_3) \end{aligned}$$

Then if the blocks are independent and have the same probability of success,  $p$ , this leads to  $P_s = 3 \cdot p^2 - 2 \cdot p^3$ .

It can be transformed into  $P_s = p^3 + 3 \cdot p^2 - 3 \cdot p^3$  and finally

$$P_s = p^3 + 3 \cdot p^2 \cdot (1-p) \quad (\text{B.14})$$

is obtained.

Formula (B.14) can be generalized to an  $m/n$  logical structure made of  $n$  identical blocks. In this case  $m$  blocks out of  $n$  are required for system success and the probability of success of the system  $P_s$  is given by the general formula:

$$P_s = \sum_{r=0}^{n-m} \binom{n}{r} \cdot p^{n-r} \cdot (1-p)^r \quad (\text{B.15})$$

Applying Formula (B.15) to the 2/4 logical structure presented in Figure 9 gives:

$$P_s = p^4 + 4 \cdot p^3 \cdot (1-p) + 6 \cdot p^2 \cdot (1-p)^2 = 3 \cdot p^4 - 8 \cdot p^3 + 6 \cdot p^2 \quad (\text{B.16})$$

An  $m/n$  system needs  $m$  blocks in up states to be in up state. Then it needs  $(n-m+1)$  blocks in down states to be in down state. Therefore an  $m/n$  system with regards to up state is a  $(n-m+1)/n$  with regards to down state and the probability of failure of an  $m/n$  system is given by exchanging  $m \leftrightarrow (n-m+1)$  and  $p \leftrightarrow (1-p)$  in Formula (B.15):

$$P_c = \sum_{r=0}^{m-1} \binom{n}{r} \cdot (1-p)^{n-r} \cdot p^r \quad (\text{B.17})$$

### Particular cases

- When  $m = n - 1$  (e.g. 2/3, 3/4, etc.) Formula (B.15) is reduced to:

$$P_s = n \cdot p^m + m \cdot p^n \quad (\text{B.18})$$

- When  $n = 2m - 1$ , the systems are symmetrical with regards to success and failures: the system is in up state if  $m$  blocks are in up states and is in down state if  $m$  blocks are in down states. This is the case for the 1/1, 2/3, 3/5, etc. logical structures. Because of this property, the 2/3 structure is widely used in industry for designing safety systems.

If the  $n$  items are not identical, use of a more general procedure is recommended (see 11.8.2).

## B.5 Use of Sylvester-Poincaré formula for large RBDs and repeated blocks

### B.5.1 General

When repeated blocks are implemented, the formulae developed in Clause B.3 can be applied only for the parts of the RBD which do not contain the repeated blocks. For other parts of the RBD, the repeated blocks shall be properly taken into account.

Equivalent RBDs made of success paths (minimal tie sets) or of failure combinations (minimal cut sets) are typical RBDs with such repeated blocks. This is why they are analysed hereafter.

### B.5.2 Sylvester-Poincaré formula with tie sets

The success state of a system having  $n$  success path (minimal tie sets),  $(I_i)$ , can be written:

$$s = \bigcup_i \Pi_i$$

The minimal tie sets,  $\Pi_i$ , are not independent from each other and the corresponding Sylvester-Poincaré formula therefore takes the following form:

$$\begin{aligned} P_s &= P\left(\bigcup_{i=1}^n \Pi_i\right) = \sum_i P(\Pi_i) - \sum_{i<j} P(\Pi_i \bullet \Pi_j) + \sum_{i<j<k} P(\Pi_i \bullet \Pi_j \bullet \Pi_k) - \text{etc.} \\ &= SP_i - SP_{ij} + SP_{ijk} - \text{etc.} \end{aligned} \quad (\text{B.19})$$

Formula (B.19) expresses that the probability of the union of the tie sets is equal to

- 1) the sum of the probability of the tie sets (term  $SP_i$ ),
- 2) minus the sum of the probabilities of the intersection of the tie sets  $2 \times 2$  (term  $SP_{ij}$ ),
- 3) plus the sum of the probabilities of the intersection of the tie sets  $3 \times 3$  (term  $SP_{ijk}$ ),
- 4) minus the sum of the probabilities of the intersection of the tie sets  $4 \times 4$  (term  $SP_{ijkl}$ ),
- 5) etc.

The tie sets are not independent as the same event can appear in several tie sets. Therefore, it is necessary to analyse all the intersections of the tie sets before doing the probabilistic calculations in order to simplify them when they include identical events.

This can be shown with the example developed in Clause 8 which comprises 4 minimal tie sets:  $\Pi_1 = b_1 \bullet c_1$ ,  $\Pi_2 = a \bullet c_1$ ,  $\Pi_3 = a \bullet c_2$ ,  $\Pi_4 = b_2 \bullet c_2$ .

Implementing Formula (B.19) leads to the following calculations.

- a) First term  $SP_i$ :

$$\sum_i P(\Pi_i) = P_{b_1} \cdot P_{c_1} + P_a \cdot P_{c_1} + P_a \cdot P_{c_2} + P_{b_2} \cdot P_{c_2}$$

- b) Second term  $SP_{ij}$ :

$$\begin{aligned} \sum_{i<j} P(\Pi_i \bullet \Pi_j) &= P(\Pi_1 \bullet \Pi_2) + P(\Pi_1 \bullet \Pi_3) + P(\Pi_1 \bullet \Pi_4) + P(\Pi_2 \bullet \Pi_3) + P(\Pi_2 \bullet \Pi_4) + P(\Pi_3 \bullet \Pi_4) \\ &= P(b_1 \bullet c_1 \bullet a \bullet c_1) + P(b_1 \bullet c_1 \bullet a \bullet c_2) + P(b_1 \bullet c_1 \bullet b_2 \bullet c_2) + P(a \bullet c_1 \bullet a \bullet c_2) + P(a \bullet c_1 \bullet b_2 \bullet c_2) + P(a \bullet c_2 \bullet b_2 \bullet c_2) \\ &= P(b_1 \bullet c_1 \bullet a) + P(b_1 \bullet c_1 \bullet a \bullet c_2) + P(b_1 \bullet c_1 \bullet b_2 \bullet c_2) + P(a \bullet c_1 \bullet c_2) + P(a \bullet c_1 \bullet b_2 \bullet c_2) + P(a \bullet b_2 \bullet c_2) \\ &= P_{b_1} \cdot P_{c_1} \cdot P_a + P_{b_1} \cdot P_{c_1} \cdot P_a \cdot P_{c_2} + P_{b_1} \cdot P_{c_1} \cdot P_{b_2} \cdot P_{c_2} + P_a \cdot P_{c_1} \cdot P_{c_2} + P_a \cdot P_{c_1} \cdot P_{b_2} \cdot P_{c_2} + P_a \cdot P_{b_2} \cdot P_{c_2} \end{aligned}$$

- c) Third term  $SP_{ijk}$ :

$$\begin{aligned} \sum_{i<j<k} P(\Pi_i \bullet \Pi_j \bullet \Pi_k) &= P(\Pi_1 \bullet \Pi_2 \bullet \Pi_3) + P(\Pi_1 \bullet \Pi_2 \bullet \Pi_4) + P(\Pi_2 \bullet \Pi_3 \bullet \Pi_4) \\ &= P(b_1 \bullet c_1 \bullet a \bullet c_1 \bullet a \bullet c_2) + P(b_1 \bullet c_1 \bullet a \bullet c_1 \bullet b_2 \bullet c_2) + P(a \bullet c_1 \bullet a \bullet c_2 \bullet b_2 \bullet c_2) \\ &= P(b_1 \bullet c_1 \bullet a \bullet c_2) + P(b_1 \bullet c_1 \bullet a \bullet b_2 \bullet c_2) + P(a \bullet c_1 \bullet b_2 \bullet c_2) \\ &= P_{b_1} \cdot P_{c_1} \cdot P_a \cdot P_{c_2} + P_{b_1} \cdot P_{c_1} \cdot P_a \cdot P_{b_2} \cdot P_{c_2} + P_a \cdot P_{c_1} \cdot P_{b_2} \cdot P_{c_2} \end{aligned}$$

- d) Fourth term  $SP_{ijkl}$ :

$$\begin{aligned} \sum_{i<j<k<l} P(\Pi_i \bullet \Pi_j \bullet \Pi_k \bullet \Pi_l) &= P(\Pi_1 \bullet \Pi_2 \bullet \Pi_3 \bullet \Pi_4) \\ &= P(b_1 \bullet c_1 \bullet a \bullet c_1 \bullet a \bullet c_2 \bullet b_2 \bullet c_2) = P(b_1 \bullet c_1 \bullet a \bullet c_2 \bullet b_2) = P_{b_1} \cdot P_{c_1} \cdot P_a \cdot P_{c_2} \cdot P_{b_2} \end{aligned}$$

Therefore, with these 4 minimal tie sets  $4 + 6 + 3 + 1 = 14$  terms to be calculated have been identified.

As the probability of tie sets,  $P(\Pi_i)$ , are normally not small compared to 1, the probabilities of  $P(\Pi_i \cdot \Pi_j)$ ,  $P(\Pi_i \cdot \Pi_j \cdot \Pi_k)$  are also not small and cannot be neglected.

With 3 events with high probabilities  $P_a = P_b = P_c = 0,9$  the following results are obtained:

$$P(a+b+c) = P_a + P_b + P_c - (P_a \cdot P_b + P_a \cdot P_c + P_b \cdot P_c) + P_a \cdot P_b \cdot P_c$$

$$P(a+b+c) = 2,7 - 2,43 + 0,729 = 0,999$$

Therefore, no term is negligible and all terms have to be considered in the calculations. Therefore Formula (B.19) is not really manageable because too many terms are needed to obtain suitable approximations.

### B.5.3 Sylvester-Poincaré formula with cut sets

The failed state of a system having  $m$  failure paths (minimal cut sets),  $(C_i)$  can be written:

$$\bar{s} = \bigcup_i C_i$$

This leads to the corresponding Sylvester-Poincaré formula:

$$\begin{aligned} P_s = 1 - P_{\bar{s}} &= P\left(\bigcup_{i=1}^m C_i\right) = \sum_i P(C_i) - \sum_{i<j} P(C_i \cdot C_j) + \sum_{i<j<k} P(C_i \cdot C_j \cdot C_k) - \text{etc.} \\ &= SP_i - SP_{ij} + SP_{ijk} - \text{etc.} \end{aligned} \quad (\text{B.20})$$

The example from Clause 8 provides also 4 minimal cut sets

$$C_1 = \bar{b}_1 \cdot \bar{a} \cdot \bar{b}_2, \quad C_2 = \bar{c}_1 \cdot \bar{c}_2, \quad C_3 = \bar{b}_1 \cdot \bar{a} \cdot \bar{c}_2, \quad C_4 = \bar{b}_2 \cdot \bar{a} \cdot \bar{c}_2$$

and, as with the tie sets, this also implies the calculations of 14 terms however the situation is very different because the probabilities  $P(C_i)$  are normally small compared to 1.

Then the probabilities of  $P(C_i \cdot C_j)$ ,  $P(C_i \cdot C_j \cdot C_k)$ , etc. are smaller and smaller and Formula (B.20) converges rather quickly. Therefore, approximations are available.

If the formula  $P(a+b+c) = P_a + P_b + P_c - (P_a \cdot P_b + P_a \cdot P_c + P_b \cdot P_c) + P_a \cdot P_b \cdot P_c$  is considered, the results with 3 events with high ( $P_a = P_b = P_c = 0,9$ ) and 3 events with low probabilities ( $P_a = P_b = P_c = 0,01$ ) can be compared:

- $P(a+b+c) = 2,7 - 2,43 + 0,729 = 0,999$  is obtained with  $P_a = P_b = P_c = 0,9$ ;
- $P(a+b+c) = 0,03 - 0,0003 + 0,000001 = 0,029701$  is obtained with  $P_a = P_b = P_c = 0,01$ .

Then in the case with low probabilities

- the term  $SP_i$  alone provides an upper bound of the probability: 0,03,
- the sum  $SP_i - SP_{ij}$  provides a lower bound: 0,0297,



and the exact result belongs to the interval  $[0,03 - 0,029\ 7]$ .

These results can be extrapolated to a large number of events:

- when the probabilities are high, the convergence is very slow and all the terms have to be considered to obtain the result;
- when the probabilities are low, the convergence is fast and the first term of the Sylvester-Poincaré formula provides an acceptable approximated result and the first two terms a good interval to which the result belongs.

Then the Sylvester-Poincaré Formula (B.20) using cut sets ( $C_i$ ) which generally involves low probabilities, is a better candidate than Formula (B.19) using tie sets ( $I_i$ ) to derive acceptable approximations. Therefore, it is better to consider the minimal cut sets ( $C_i$ ) than the tie sets ( $I_i$ ) when performing probabilistic calculations.

The lower the resulting probability,  $P_s$ , the faster the convergence is obtained and in the best cases the following approximation works well:

$$P_s = P\left(\bigcup_{i=1}^n C_i\right) \approx \sum_i P_{C_i} \quad (\text{B.21})$$

This approximation is widely used and is the basis of calculations performed by numerous software packages available for availability/reliability calculations on RBD or fault trees. In some cases the second term of the Sylvester-Poincaré formula is calculated in order to provide the bounds of the interval framing  $P_s$ .

## B.6 Method for disjointing Boolean expressions

### B.6.1 General and background

Formula (B.7) can be written in the equivalent form:

$$s = a + b \equiv (a + b) \bullet \Omega = (a + b) \bullet (a + \bar{a}) = a + \bar{a} \bullet b \quad (\text{B.22})$$

The process of rewriting Formula (B.7) in the form of Formula (B.22) will be referred to as *disjointing*.

In Formula (B.22) the terms  $a$  and  $\bar{a} \bullet b$  are disjoint terms. This implies that  $a \bullet (\bar{a} \bullet b) = \Phi$  and therefore  $P[a \bullet (\bar{a} \bullet b)] = 0$ . Then  $P_s$  is reduced to the well-known result:

$$P_s = P_a + (1 - P_a) \cdot P_b \quad (\text{B.23})$$

Note that it is also possible to write Formula (B.7) in other disjointed forms, one of which is  $s = b + \bar{b} \bullet a$  leading to another correct expression, namely:

$$P_s = P_b + (1 - P_b) \cdot P_a \quad (\text{B.24})$$

Needless to say Formulae (B.23) and (B.24) are equivalent to the formulae  $P_s = P_a + P_b - P_a \cdot P_b$  and  $P_s = 1 - (1 - P_a) \cdot (1 - P_b)$  previously found.

The difference with the previous calculations is that the number of terms in the probability formula is the same as the number of disjoint terms in the Boolean equation.

Let us consider that  $s$  is expressed by the union of disjoint success paths:

$$s = \bigcup_i \Pi_i^d \text{ where } \Pi_i^d \cap \Pi_j^d = \Phi \forall i, j.$$

Then the Sylvester-Poincaré formula will be reduced to:

$$P_s = P\left(\bigcup_{i=1}^n \Pi_i^d\right) = \sum_i P_{\Pi_i^d} \quad (\text{B.25})$$

In the same way if  $\bar{s}$  is expressed by the union of disjoint cut sets the following formula is obtained:

$$\bar{s} = \bigcup_i C_i^d \text{ where } C_i^d \cap C_j^d = \Phi \forall i, j$$

and the Sylvester-Poincaré formula will be reduced to:

$$P_{\bar{s}} = P\left(\bigcup_{i=1}^n C_i^d\right) = \sum_i P_{C_i^d} \quad (\text{B.26})$$

Therefore and provided the minimal tie sets or cut sets are disjoint, Formulae (B.25) and (B.26) can be used to perform exact calculations. When the block failure probabilities vary with time they can be used for availability,  $A_S(t)$ , or unavailability  $U_S(t)$  calculations as well.

The primary objective therefore is to be able to cast Boolean expressions for system success (or to system failure) into a disjointed form. This means that each term in the final Boolean expression is disjoint with respect to every other term. Further details of the method can be found in [19].

It should be noted that two terms are mutually disjoint if at least one variable in one term appears in its complementary form in the other. For example the terms  $p \bullet q \bullet r \bullet s$  and  $\bar{s} \bullet t \bullet u$  are disjoint by virtue of  $s$ . The converse is also true. Namely two terms are not disjoint (i.e., they overlap) if none of the variables in one term appear in complementary form in the other. For example, the two terms  $p \bullet q \bullet r$  and  $s \bullet t \bullet u$  are not mutually disjoint.

### B.6.2 Disjointing principle

If two terms  $\theta_1$  and  $\theta_2$  are not disjoint, and it is required to make  $\theta_2$  disjoint with respect to  $\theta_1$  several procedures may be used.

The basic principle is the following:

- pick out all the variables in  $\theta_1$  which do not appear in  $\theta_2$  (such terms are known collectively as the relative complement of  $\theta_2$  with respect to  $\theta_1$ ). Suppose the relative complement is  $v_1 \bullet v_2 \bullet v_3 \bullet v_4$ ;

- then replace  $\theta_2$  by

$$\theta_2^* = \bar{v}_1 \bullet \theta_2 + v_1 \bullet \bar{v}_2 \bullet \theta_2 + v_1 \bullet v_2 \bullet \bar{v}_3 \bullet \theta_2 + v_1 \bullet v_2 \bullet v_3 \bullet \bar{v}_4 \bullet \theta_2.$$

The resulting expression  $\bar{\theta}_1 + \theta_2^*$  will consist of terms which will all be disjoint with respect to one another.

For example, to make the term  $\theta_2 = d \bullet e \bullet f$  disjoint with respect to the term  $\theta_1 = a \bullet b \bullet c \bullet d \bullet e$ , proceed as follows:

- the relative complement of  $\theta_2$  with respect to  $\theta_1$  is  $a \bullet b \bullet c$ ;
- so that if  $\theta_2$  is replaced by

$$\theta_2^* = \bar{a} \bullet d \bullet e \bullet f + a \bullet \bar{b} \bullet d \bullet e \bullet f + a \bullet b \bullet \bar{c} \bullet d \bullet e \bullet f$$

then  $\theta_1$  and  $\theta_2^*$  will be disjoint with respect to one another.

### B.6.3 Disjointing procedure

The basic disjointing procedure is as follows:

- a) express system success (denoted by  $s_1$ ) in “sum-of-product” Boolean terms<sup>3</sup> (i.e. tie sets) and label the terms from left to right, “ $\theta_{11}, \theta_{12}, \theta_{13}, \dots$ ”;
- b) select  $\theta_{11}$  as a “pivotal” term and compare  $\theta_{12}$  with  $\theta_{11}$ ;
- c) if necessary (i.e. if the two terms are not disjoint), make  $\theta_{12}$  disjoint with respect to  $\theta_{11}$  as described in B.6.2;
- d) if necessary, make  $\theta_{13}$  disjoint with respect to  $\theta_{11}$ ;
- e) continue the process for the remaining terms in  $s_1$ ;
- f) examine the somewhat expanded (on account of additional terms added) expression reached at this stage, and simplify (where possible) using the rules of Boolean algebra (make use of rules such as  $x+x=x$ ,  $x+x \bullet y=x$ ,  $x \bullet y + \bar{x} \bullet y=y$ ). Call the resulting expression  $s_2$  and label the terms from left to right, “ $\theta_{21}, \theta_{22}, \theta_{23}, \dots$ ”;
- g) select the second term ( $\theta_{22}$ ) of  $s_2$  as a “pivotal” term and compare  $\theta_{23}$  with  $\theta_{22}$ , and proceed as indicated in c) to f) but using the terms of  $s_2$ . Call the resulting expression  $s_3$ ;
- h) continue as above until all the terms have been used as “pivotal” terms by which time the final expression obtained will be the fully disjointed version of the original expression  $s_1$ .

Finally a set of disjoint terms ( $\Pi_i^d$ ) related to the success of the system as described in B.6.1 is obtained. Therefore, the probability of success  $P_S$  or the availability  $A_S(t)$  of the system can be calculated by applying Formula (B.25).

The same procedure may be used to obtain the disjoint terms ( $C_i^d$ ) related to the failure of the system as described in B.6.1. Therefore, the probability of failure  $P_S^-$  or the unavailability  $U_S(t)$  of the system can be calculated by applying Formula (B.26).

The procedure described is very basic and can be improved as this is done to process the example given in B.6.4.

### B.6.4 Example of application of disjointing procedure

It is supposed that a network or system consists of five elements A, B, C, D and E and that  $a, b, c, d$  and  $e$  denote the corresponding Boolean “success” variables. It is also supposed that

<sup>3</sup> For particularly simple Boolean expressions for system success, single as well as products of two or more terms may be used.

system success in Boolean terms ( $s$ ) is defined by the following expression, which comprises four sum-of-product terms (i.e. tie sets):

$$s = a \bullet b + e \bullet b + e \bullet d + c \bullet d$$

To make the above expression disjoint, the basic procedure described in B.6.3 can be improved and applied as follows:

**Step 0** – Classification of the paths by increasing lengths and in alphabetic order

$$s = a \bullet b + b \bullet e + c \bullet d + d \bullet e$$

**Step 1** – The disjunctive procedure starts with the last product ( $d \bullet e$ ) to make it disjoint from all its predecessors:

- **1.1:** This procedure applies to its immediate predecessor ( $c \bullet d$ ) by:
  - identifying the event(s) belonging to  $c \bullet d$  but not to  $d \bullet e$ . This gives  $c$ ;
  - changing  $d \bullet e$  by  $d \bullet e \bullet \bar{c}$  in the original formula.
- **1.2:** Reiterate step 1.1 with the next term on the left ( $b \bullet e$ ) by:
  - identifying the event(s) belonging to  $b \bullet e$  but not to  $d \bullet e$ . This gives  $b$ ;
  - changing the expression of  $d \bullet e$  modified in step 1.1, (i.e.,  $d \bullet e \bullet \bar{c}$ ), by  $d \bullet e \bullet \bar{c} \bullet \bar{b}$  in the original formula.
- **1.3:** It is not necessary to reiterate step 1.2 with the next term on the left ( $a \bullet b$ ), because the last product  $d \bullet e$  modified as  $d \bullet e \bullet \bar{c} \bullet \bar{b}$  is already made disjoint from  $a \bullet b$ . The disjunctive procedure applied to  $d \bullet e$  is now achieved.

The original formula can be rewritten as follows:  $s = a \bullet b + b \bullet e + c \bullet d + d \bullet e \bullet \bar{c} \bullet \bar{b}$ .

**Step 2** – Reiterate the above procedure (steps 1.1 to 1.3) to make  $c \bullet d$  disjoint from its predecessors.

- **2.1:** As previously the procedure is first applied to its immediate predecessor  $b \bullet e$  by:
  - identifying the event(s) belonging to  $b \bullet e$  but not to  $c \bullet d$ . This gives  $b$  and  $e$ ;
  - changing the expression of  $c \bullet d$  by  $c \bullet d \bullet \overline{b \bullet e}$  in the original formula, by keeping in mind (De Morgan law) that  $c \bullet d \bullet \overline{b \bullet e} = c \bullet d \bullet (\bar{b} + \bar{e}) = c \bullet d \bullet (\bar{b} + b \bullet \bar{e})$ . It becomes:

$$s = a \bullet b + b \bullet e + c \bullet d \bullet \bar{b} + c \bullet d \bullet b \bullet \bar{e} + d \bullet e \bullet \bar{c} \bullet \bar{b}$$

- **2.2:** Because the first term of the decomposition of  $c \bullet d$ , ( i.e.,  $c \bullet d \bullet \bar{b}$  is already disjoint from all its predecessors  $a \bullet b$  and  $b \bullet e$ , and the second term  $c \bullet d \bullet b \bullet \bar{e}$  is already disjoint from its predecessor  $b \bullet e$ , it remains to make  $c \bullet d$  disjoint from its second predecessor  $a \bullet b$  by:
  - identifying the event(s) belonging to  $a \bullet b$  but not to  $c \bullet d \bullet b \bullet \bar{e}$ . This gives  $a$ ;
  - changing the expression of  $c \bullet d \bullet b \bullet \bar{e}$  by  $c \bullet d \bullet b \bullet \bar{e} \bullet \bar{a}$  in the original formula that becomes:

$$s = a \bullet b + b \bullet e + c \bullet d \bullet \bar{b} + c \bullet d \bullet b \bullet \bar{e} \bullet \bar{a} + d \bullet e \bullet \bar{c} \bullet \bar{b}$$

**Step 3** – Reiterate the disjunctive procedure to make  $b \bullet e$  disjoint from its unique predecessor  $a \bullet b$  by:

- identifying the event(s) belonging to  $a \bullet b$  but not to  $b \bullet e$ . This gives  $a$ ;
- changing the expression of  $b \bullet e$  by  $b \bullet e \bullet \bar{a}$  in the original formula.

**Step 4** – Because the first product (here  $a \bullet b$ ) remains always unchanged, the procedure is achieved and gives the following final sum of disjoint products:

$$s = a \bullet b + b \bullet e \bullet \bar{a} + c \bullet d \bullet \bar{b} + c \bullet d \bullet b \bullet \bar{e} \bullet \bar{a} + d \bullet e \bullet \bar{c} \bullet \bar{b}$$

Finally 5 disjoint terms are obtained

- $\Pi_1^d = a \bullet b$ ,
- $\Pi_2^d = b \bullet e \bullet \bar{a}$ ,
- $\Pi_3^d = c \bullet d \bullet \bar{b}$ ,
- $\Pi_4^d = c \bullet d \bullet b \bullet \bar{e} \bullet \bar{a}$ ,
- $\Pi_5^d = d \bullet e \bullet \bar{c} \bullet \bar{b}$ ,

and  $s$  can be written  $s = \bigcup_i \Pi_i^d$  as described in B.6.1.

Therefore, the probability of success  $P_s$  or the availability  $A_s(t)$  of the system can be calculated by applying Formula (B.25):  $P_s = P(\bigcup_{i=1}^n \Pi_i^d) = \sum_i P_{\Pi_i^d}$ .

Finally the system availability is found as:

$$A_S(t) = A_A(t) \cdot A_B(t) + A_B(t) \cdot A_E(t) \cdot [1 - A_A(t)] + A_C(t) \cdot A_D(t) \cdot [1 - A_B(t)] + A_C(t) \cdot A_D(t) \cdot A_B(t) \cdot [1 - A_E(t)] \cdot [1 - A_A(t)] + A_D(t) \cdot A_E(t) \cdot [1 - A_C(t)] \cdot [1 - A_B(t)]$$

The number of disjoint terms depends on the order in which the success paths are used to apply the disjunction algorithms. All results are equivalent but are obtained more or less quickly. There is no theoretical optimum and the choice can be based on heuristics which have proven to work well. The use of the alphabetical order is an example of such a heuristic.

Of course the same procedure can be used with the minimal cut sets to find disjoint sets ( $C_i^d$ ) allowing to calculate the probability of failure  $P_s$  or the unavailability  $U_s(t)$  of the system by

applying Formula (B.26):  $P_s = P(\bigcup_{i=1}^n C_i^d) = \sum_i P_{C_i^d}$ .

### B.6.5 Comments

The most important attribute of the procedures described in B.6.4 is that the sequence of steps needed to carry out the disjointing is relatively straightforward to program for running on a computer. The improved procedure described in B.6.4 is often used on modern PCs where quite complicated sum-of-product Boolean expressions can be disjointed almost instantaneously. It is intended that the details given in this standard will be sufficient to enable a suitable program to be written.

Another important attribute is the fact that the procedure, being primarily aimed at disjointing Boolean expressions, can be applied with equal efficacy to Boolean expressions arising from fault tree analyses.

## B.7 Binary decision diagrams

### B.7.1 Establishing a BDD

At the present time the state of the art in probabilistic calculations on Boolean functions is the use of the Shannon decomposition of the Boolean functions in order to build binary decision diagrams (BDD) encoding all the disjoint combinations leading to the realization of the function modelled by this function.

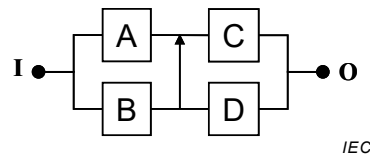


Figure B.6 – Reminder of the RBD in Figure 35

The Boolean function modelled by the RBD in Figure B.6 depends on 4 Boolean variables  $a$ ,  $b$ ,  $c$  and  $d$ .

The Shannon decomposition is similar to the truth table of the Boolean function modelled by the RBD.

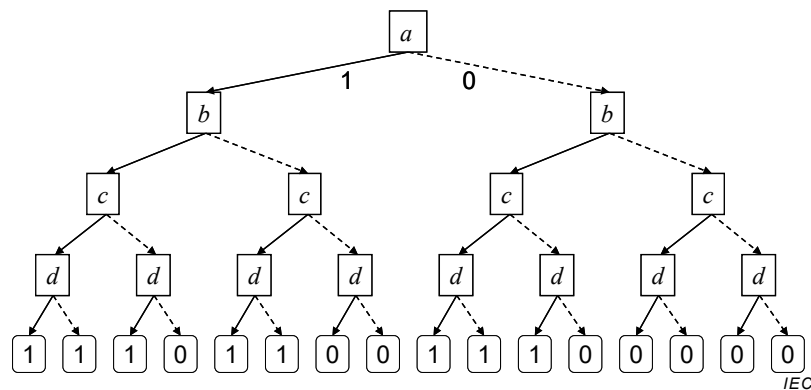


Figure B.7 – Shannon decomposition of the Boolean function represented by Figure B.6

This decomposition has been represented in a graphical form, illustrated in Figure B.7. The process to do that is the following one:

- 1) choose one of the variables (e.g.  $a$ ) and place it at the top of the graph;
- 2) from this variable, draw two arrows to represent its two possible states: e.g. 1 on the left and 0 on the right (solid and dotted lines have been used to make the figure clearer);
- 3) choose another variable (e.g.  $b$ ) and connect it to the previous arrows. This variable will appear twice;
- 4) for each occurrence of this variable draw two arrows to represent its two possible states;
- 5) choose another variable (e.g.  $c$ ) and connect it to the previous arrows. This variable will appear four times;
- 6) etc. continue the process until all variables have been processed.

Then, for  $n$  variables,  $2^n$  paths are obtained. Each of them leads either to the success of the function ( $s = 1$ ) or its failure  $\bar{s} = 0$ . This can be achieved by analysing the RBD corresponding to this Shannon decomposition.

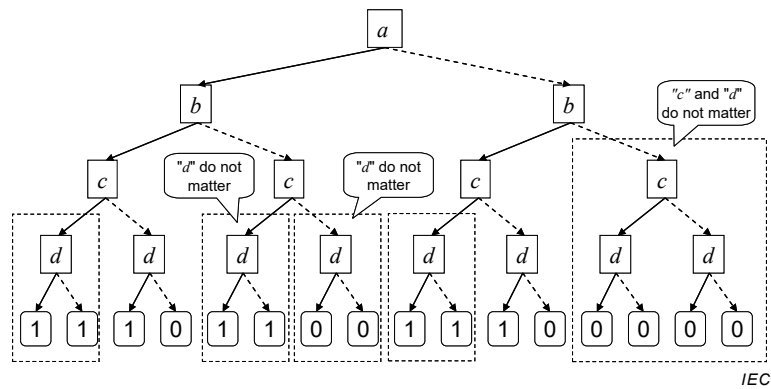


Figure B.8 – Identification of the parts which do not matter

The next step is the simplification of this graph by identifying the parts which do not matter. They have been boxed in dotted line in Figure B.8. For example on the left hand side the state of the system does not depend on the state of the variable *d* and on the right hand side it does not depend on the states of the variables *c* and *d*.

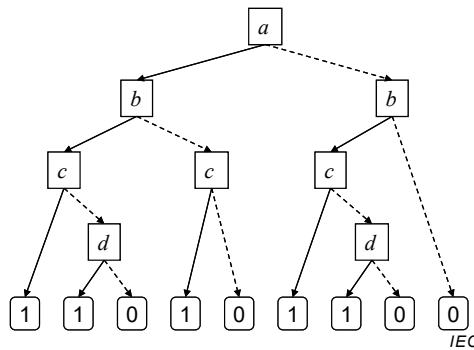


Figure B.9 – Simplification of the Shannon decomposition

This allows to obtain the simplified the graph shown in Figure B.9. It is not identical to the graph previously obtained in Figure 38 for the same RBD. This is due to a different choice for the order of the variables used for the decomposition. This shows that the decomposition is not unique and leads to a more or less simple graph according to the order which has been chosen.

On this graph 9 paths can be identified: 5 paths lead to the success of *S* and 4 to its failure. Reading those paths from the BDD allows to obtain the relationship between *s* or  $\bar{s}$  and the states of the variables *a*, *b*, *c* and *d*:

$$s = a \bullet b \bullet c + a \bullet b \bullet \bar{c} \bullet d + a \bullet \bar{b} \bullet c + a \bullet \bar{b} \bullet \bar{c} \bullet d + a \bullet b \bullet \bar{c} \bullet d$$

$$\bar{s} = a \bullet b \bullet \bar{c} \bullet \bar{d} + a \bullet \bar{b} \bullet \bar{c} + a \bullet b \bullet \bar{c} \bullet \bar{d} + a \bullet b$$

The next step is to build the BDD related to this RBD. As shown in Figure B.10 this is done just by gathering the inputs with the same values.

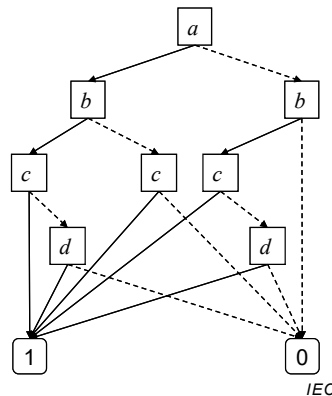


Figure B.10 – Binary decision diagram related to the RBD in Figure B.6

**B.7.2 Minimal success paths and cut sets with BDDs**

The same block can appear in different places but in the basic RBD, it has always the same state in its various locations. This implies that the RBD is "coherent" and that the related Boolean function is "monotonic". That means that if the system is failed it cannot be repaired by a further block failure or if the system is in the up state it cannot be failed by a further block repair. In this case the Boolean function can be represented by the union of the minimal tie sets (success paths) and its complementary function can be represented by the union of the minimal cut sets (failure paths).

When a Boolean function is non-monotonic then the concepts of minimal tie or cut sets are not relevant and must be replaced by the concept of "prime implicants". The difference is that a minimal tie set is made only of a combination of blocks in up states (and a minimal cut set only of a combination of blocks in down states), whereas a prime implicant may be made of a combination of blocks in up and down states. The prime implicants cannot be reduced to minimal tie or cut sets and should not be mixed up with the disjoint terms analysed in B.7.1 which are equivalent to a union of minimal tie or cut sets.

Therefore, if the Boolean function is monotonic and  $\Pi_i$  a success path containing failed blocks, removing the failed blocks provide also a success path. For example,  $a \bullet b \bullet c \bullet d$  being a success path,  $b \bullet d$  is also a success path.

In the same way, if  $C_i$  is a cut set containing blocks in the up state, then removing those blocks provides also a cut set: for example  $a \bullet b \bullet c \bullet \bar{d}$  being a cut set,  $c \bullet \bar{d}$  is also a cut set.

Therefore, disjoint success paths identified in B.7.1 can be used to identify the success paths of the related to the Boolean function. This is illustrated in Figure B.11.

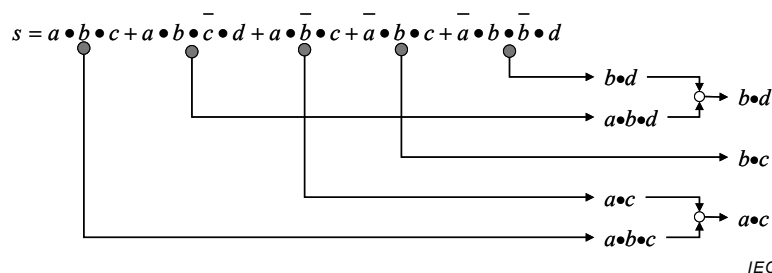
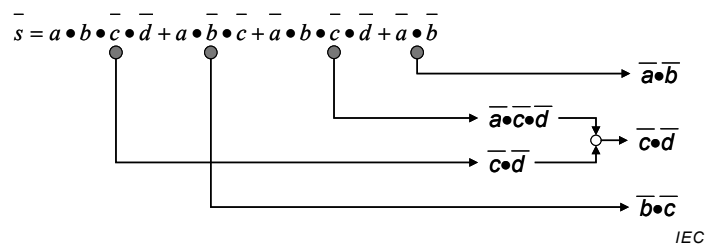


Figure B.11 – Obtaining success paths (tie sets) from an RBD



Among the found tie sets, some are non-minimal sets which are included (from the Boolean algebra point of view) in the minimal tie sets. Finally three minimal success paths are found and they are similar to those previously identified.



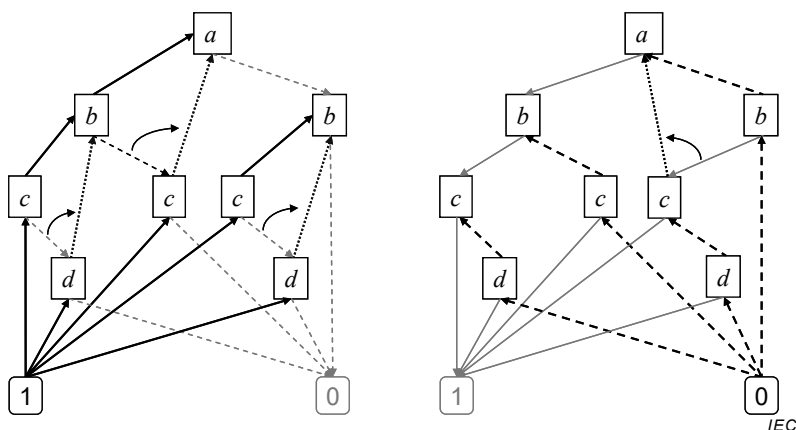
**Figure B.12 – Obtaining failure paths (cut sets) from an RBD**

Figure B.12 shows how the minimal cut sets can be found from the Boolean equation of the system failure. The principle is exactly the same as for the success paths. Three minimal cut sets are found and they are identical to those previously found.

The BDDs provide also an efficient tool to identify the minimal tie or cut sets.

The principle to find the tie sets (success paths) is illustrated in the left hand side of Figure B.13. This is valid only when the BDD is related to a "coherent" RBD as explained in B.7.2 .

The process consists in starting from the success state of the system and exploring the graph from bottom to up in the reverse order (of the variables) used to build the BDD. When exploring a branch, if a variable is found in the failed state, then it is by-passed and a new link with the variable just above is introduced. And so on. If the graph on the left hand side is considered, the next tie sets:  $(c \bullet b \bullet a)$ ,  $(d \bullet b \bullet a)$ ,  $(c \bullet a)$ ,  $(c \bullet a)$ ,  $(c \bullet b)$ ,  $(d \bullet b)$  are found. Some of these combinations are not minimal and one combination appears twice.



**Figure B.13 – Finding cut and tie sets from BDDs**

Similarly, the principle to find the cut sets (failure paths) is illustrated in the right hand side of Figure B.13. The process consists in starting from the failed state of the system and exploring the graph from bottom to up in the reverse order (of the variables) used to build the BDD. When exploring a branch, if a variable is found in the up state, then it is by-passed and a new link with the variable just above is introduced. And so on. If the graph on the right hand side is considered, the next cut sets:  $(d \bullet c)$ ,  $(c \bullet b)$ ,  $(d \bullet c \bullet a)$ ,  $(b \bullet a)$  are found. One of these combinations,  $(d \bullet c \bullet a)$ , is not minimal.

Therefore, the graphs can be simplified in order to encode only minimal combinations. This is not really easy by hand but powerful algorithms have been developed to do that on large BDDs in order to handle RBDs with millions of minimal tie or cut sets.

When the Boolean functions are non-monotonic, the minimal tie or cut sets are meaningless and must be replaced by the prime implicants. This is more complicated to handle but powerful algorithms are also available to deal with this problem.

**B.7.3 Probabilistic calculations with BDDs**

**B.7.3.1 General**

The BDD structure presented in Figure B.10 models in a very compact way all the paths leading to the system failure and to the system success. Several equivalent BDDs may be developed for the same RBD. As for the simplified Shannon decomposition, the size of those BDDs depends on the choice of the order of the variables.

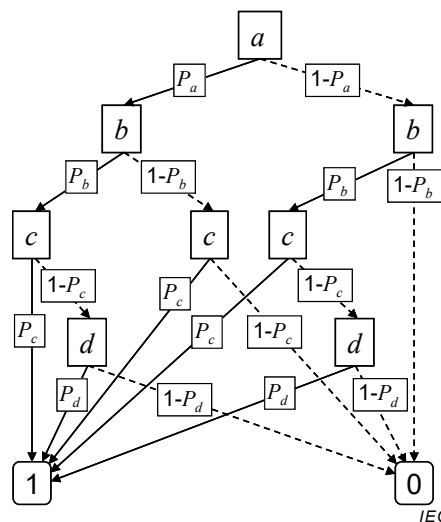
All the paths encoded within a BDD being disjoint, the BDD can be used directly for probabilistic calculations just by replacing the state variables by the corresponding probabilities of success or probabilities of failure (see Figure B.14).

Using the paths leading to the system success gives directly:

$$P_s = P_a \cdot P_b \cdot P_c + P_a \cdot P_b \cdot (1 - P_c) \cdot P_d + P_a \cdot (1 - P_b) \cdot P_c + (1 - P_a) \cdot P_b \cdot P_c + (1 - P_a) \cdot P_b \cdot (1 - P_c) \cdot P_d$$

Using the paths leading to the system failure gives directly:

$$P_s = P_a \cdot P_b \cdot (1 - P_c) \cdot (1 - P_d) + P_a \cdot (1 - P_b) \cdot (1 - P_c) + (1 - P_a) \cdot P_b \cdot (1 - P_c) \cdot (1 - P_d) + (1 - P_a) \cdot (1 - P_b)$$



**Figure B.14 – Probabilistic calculations from a BDD**

**B.7.3.2 Conditional probability calculations with RBD**

BDDs can be used to calculate conditional probabilities. Figure B.15 shows how to calculate  $P_{s|b}$  on the left hand side and  $P_{s|b}^-$  on the right hand side. This is the basis of the calculations related to conditional failure intensity (Vesely failure rate), unconditional failure intensity (failure frequency) and various importance factors (see Annex D).

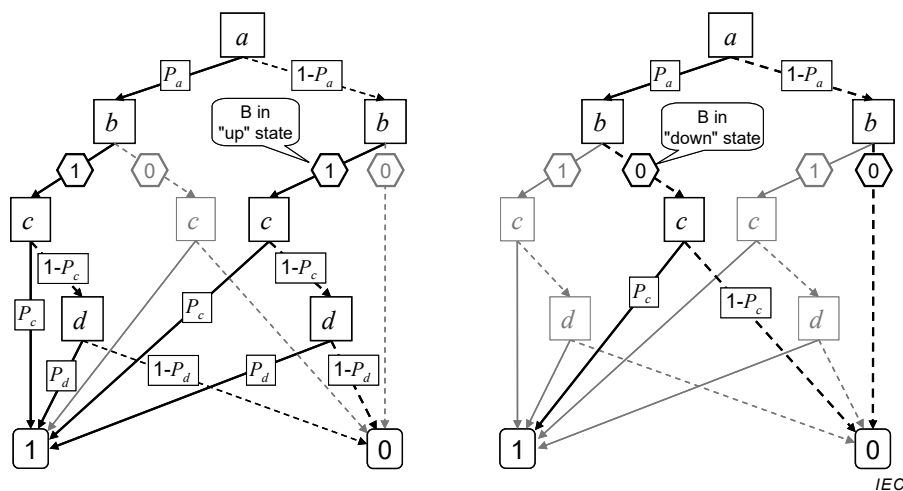


Figure B.15 – Calculation of conditional probabilities using BDDs

#### B.7.4 Key remarks about the use of BDDs

This BDD structure has proven to be very powerful to encode in a very effective and compact way all the disjoint paths leading to the system failure and the system success. This allows to perform probabilistic calculations without approximations.

The BDD can also be used to encode the minimal tie sets (success paths) or minimal cut sets (failure paths) when the RBDs are coherent or to encode the prime implicants when the RBDs are not coherent.

With  $n$  variables, the Shannon decomposition (as well as the truth table) leads to  $2^n$  paths. This is not tractable when  $n$  is large. This is why the modern BDD algorithms have been developed to build the BDD without having to build the whole Shannon decomposition. This allows to handle hundreds of variables (i.e. RBD with hundreds of blocks) and billions of success paths or minimal cut sets. The size is dependent on the choice of the order of the variables used to develop the BDDs and heuristics are available to select, to some extent, the better ones.

Using BDDs is a very effective way to store the RBD within a computer memory and to make probabilistic calculations on Boolean functions (e.g. RBDs and fault trees).

## Annex C (informative)

### Time dependent probabilities and RBD driven Markov processes

#### C.1 General

The underlying mathematics behind RBDs is Boolean algebra which is static in nature. Therefore, the probabilistic calculations with RBDs are primarily related to constant values. Nevertheless, when the blocks behave independently from each other over time, the use of the formulae developed for constant probability values can be used straightforwardly for the calculation of the system availability  $A_S(t) = P_S(t)$  from availabilities  $A_{X_i}(t) = P_{X_i}(t)$  of the blocks  $X_i$ .

The calculations can also be extended to average availability  $A_S^{\text{avg}}(t_1, t_2)$ , steady state availability  $A_S^{\text{st}}$ , asymptotic availability  $A_S^{\text{as}}$ , failure frequency  $w_S(t)$  and, only in particular cases, reliability  $R_S(t)$ .

The user of RBDs should understand that RBDs are rather more focused on availability calculations than on reliability calculations.

#### C.2 Principle for calculation of time dependent availabilities

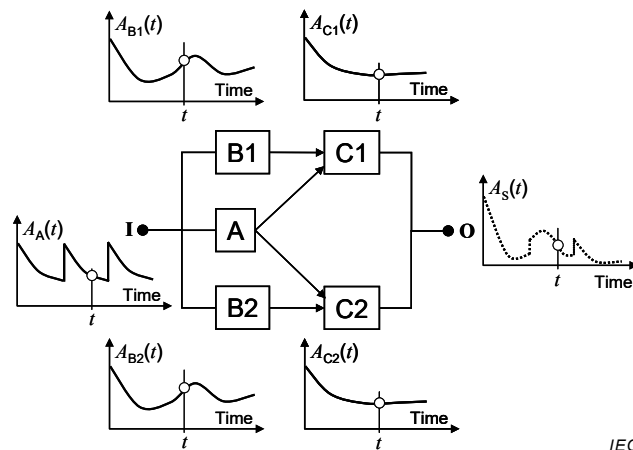


Figure C.1 – Principle of time dependent availability calculations

Figure C.1 illustrates the principle of the availability calculations by using RBDs. On this figure the availabilities of each block have been drawn. Those availabilities may have any form. The only constraint is that, according to the basic independency requirement in 5.2 d), they behave independently from each other.

Therefore, the principle is to pick up a set of block availabilities for a given time  $t$  (small circles on the figure) and use it to calculate the system availability at time  $t$  through the logic modelled by the RBD. This procedure can be used for any time in order to provide the whole evolution of the system availability  $A_S(t)$  (in dotted line on the figure).

Except in very simple cases, this procedure is not easy to handle by hand but this can be easily undertaken by the fast algorithms implemented nowadays in the RBD/FT software packages (e.g., the algorithms based on binary decision diagrams).

### C.3 Non-repaired blocks

#### C.3.1 General

The principle explained in C.2 is very easy to apply with non-repaired blocks.

#### C.3.2 Simple non-repaired blocks

For example, if the RBD presented in Figure C.1 is made of simple non-repaired blocks  $X_i$  with constant failure rates, the input curves will simply be of the classical form:

$$A_{X_i}(t) = R_{X_i}(t) = \exp(-\lambda_{X_i} \cdot t).$$

#### C.3.3 Composite block: example on a non-repaired standby system

This can also be applied to the composite block like that represented by the diagrams in Figure 11 and which models a frequently used form of redundancy known as standby redundancy (see 7.5.3 and first paragraph of Annex A).

In its most elementary form the blocks A and B are not repaired and do not behave independently from each other: B starts when A fails. Then the composite block C has to be considered as a whole (see Figure 12) and its availability  $A_C(t)$  has to be established as shown in C.2. When A and B are not repaired, C is also not repaired and therefore  $A_C(t) = R_C(t)$ .

The availability  $A_C(t)$ , of such a system can be obtained by considering what possible events may occur during a mission time  $t$ . The following are possibilities:

- block A is in up state throughout time  $t$ ; or
- block A fails at time  $\tau < t$ , item B starts at  $\tau$  (i.e., B has not failed in dormant state and the switch has not failed before  $\tau$ ) and does not fail over the interval  $[\tau, t]$ .

If it is noted:

- $\lambda_A$  is the failure rate of block A and  $f_A(\tau)$  is its failure density function;
- $\lambda_{Bd}$  is the failure rate of block B when in the passive (dormant) state, either cold or under low power;
- $\lambda_B$  is the failure rate of block B when in active state, after it has started due to the failure of A;
- $\lambda_{SW}$  is the failure rate of the switch S and  $R_{SW}(\tau)$  is its reliability at time  $\tau$ .

This leads to the following mathematical expression:

$$A_C(t) \equiv R_C(t) = R_A(t) + \int_0^t f_A(\tau) \cdot R_{Bd}(\tau) \cdot R_{SW}(\tau) \cdot R_B(t - \tau) \cdot d\tau$$

If it is assumed that all items have a constant active or dormant failure rate, then this mathematical expression becomes:

$$A_C(t) \equiv R_C(t) = e^{-\lambda_A t} + \int_0^t \lambda_A \cdot e^{-\lambda_A \tau} \cdot e^{-\lambda_{Bd} \tau} e^{-\lambda_{SW} \tau} e^{-\lambda_B \cdot (t-\tau)} \cdot d\tau$$

NOTE If the reliability of the switch is not a function of time but a function of some other variable (number of operations, demands, etc.), it would be preferable not to use functional notation at all, but to use instead  $P_{sw}$  to denote the switch reliability or  $\gamma_B$  to denote the probability of B to fail to start on demand.

On evaluating the integral of the above mathematical expression:

$$A_C(t) \equiv R_C(t) = e^{-\lambda_A t} + \frac{\lambda_A}{\lambda_A + \lambda_{SW} + \lambda_{Bd} - \lambda_B} \cdot \left[ e^{-\lambda_B t} - e^{-(\lambda_A + \lambda_{SW} + \lambda_{Bd})t} \right]$$

With an assumption of perfect switching,  $\lambda_{SW} = 0$ , the equation becomes:

$$A_C(t) \equiv R_C(t) = e^{-\lambda_A t} + \frac{\lambda_A}{\lambda_A + \lambda_{Bd} - \lambda_B} \cdot \left[ e^{-\lambda_B t} - e^{-(\lambda_A + \lambda_{Bd})t} \right]$$

If the dormant failure rate of item B is also assumed equal to zero, then the availability of a standby redundant system is:

$$A_C(t) \equiv R_C(t) = e^{-\lambda_A t} + \frac{\lambda_A}{\lambda_A - \lambda_B} \cdot \left[ e^{-\lambda_B t} - e^{-\lambda_A t} \right] \quad (C.1)$$

If, in addition, both failure rates are equal ( $\lambda_A = \lambda$  and  $\lambda_B = \lambda$ ), then the formula for system availability can be shown to be given by:

$$A_C(t) \equiv R_C(t) = e^{-\lambda t} \cdot (1 + \lambda \cdot t) \quad (C.2)$$

If, under such ideal conditions, there are  $n$  (instead of one) items on standby, this latter formula becomes:

$$A_C(t) \equiv R_C(t) = e^{-\lambda t} \left( 1 + \lambda \cdot t + \frac{(\lambda \cdot t)^2}{2!} + \frac{(\lambda \cdot t)^3}{3!} + \dots + \frac{(\lambda \cdot t)^n}{n!} \right) \quad (C.3)$$

It should be noted that a practical RBD should include blocks to represent the availability of the switch plus sensing mechanism, which is often the "weak link" in standby systems.

Formulae (C.1), (C.2) and (C.3) can be used for the composite block C in the same way the ordinary formulae are used for ordinary blocks. Nevertheless, establishing those formulae is difficult and other procedures, such as Markov analysis, should be used to analyse standby systems.

### C.4 RBD driven Markov processes

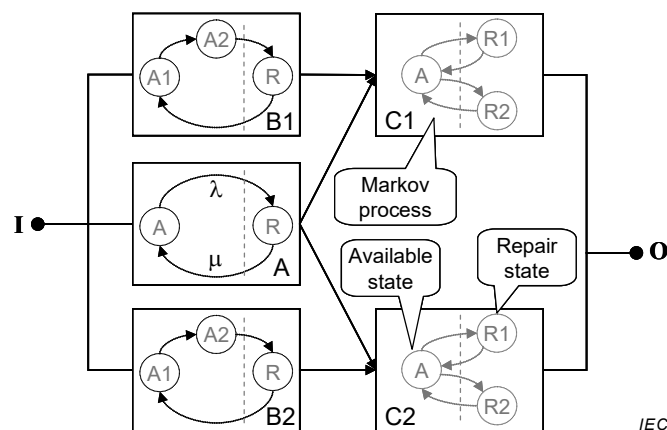


Figure C.2 – Principle of RBD driven Markov processes

As said in the previous clauses, the block availabilities may have any form and, as shown in Figure C.2, they can be calculated by using Markov processes. Such a model which is a mix between RBD and Markov graphs is an "RBD driven Markov process": the RBD provides the backbone of the model and the small Markov graphs the availabilities of the blocks. It is a way to build Markov processes for large systems and help to avoid the combinatorial explosion of the number of states.

This approach covers most of the problems encountered when dealing with repaired blocks as in most of the cases only constant failure and repair rates are considered.

In Figure C.2 the block availabilities are modelled by single Markov graphs where the repairs start as soon as the failures occur. Then, after a transient period, asymptotic values are reached and this leads to the typical behaviour of the availability illustrated in Figure C.3.

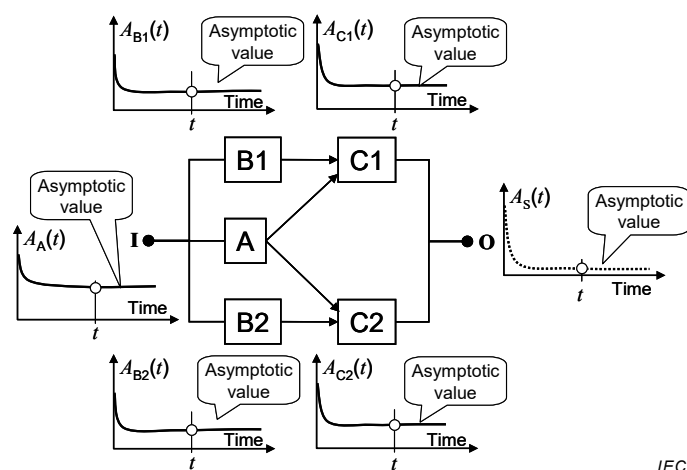
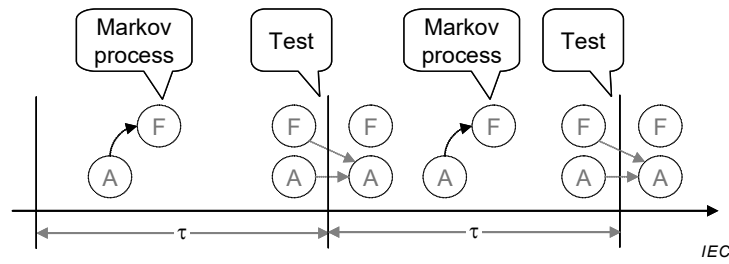


Figure C.3 – Typical availability of RBD with quickly repaired failures

For example, the availability of the block A which is modelled by the parameters  $(\lambda, \mu)$ , reaches an asymptotic value  $A_A^{as} = \frac{\mu}{\lambda + \mu}$  after a duration equal to 2 or 3 MTTRs (where  $MTTR = 1/\mu$ ).

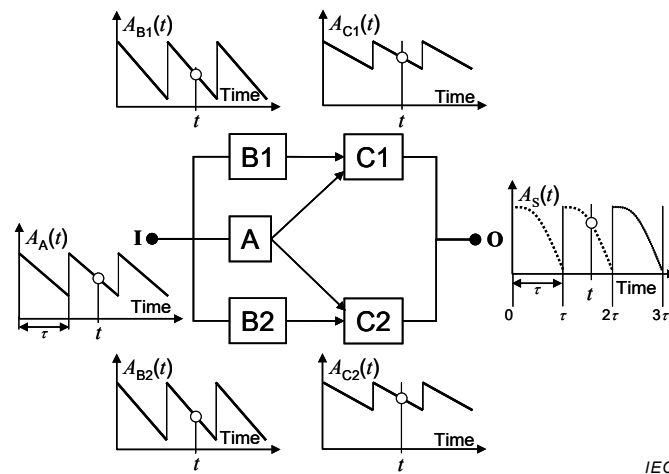
The RBD driven Markov process can also be implemented when the blocks have hidden failures which are not immediately detected when they occur. In this case, periodic tests have

to be performed to detect the failures and repair them. This cannot be modelled by single Markov graphs as in Figure C.2 but "multi-phase" Markov processes have to be used instead.



**Figure C.4 – Example of simple multi-phase Markov process**

Figure C.4 illustrates a simple multi-phase Markov model related to a periodically tested block: each test interval is a phase and the probabilities of states at the beginning of a phase are calculated from the probabilities at the end of the previous one. Then, during the test intervals, the block availability is modelled by a simple Markov graph where failure F is hidden and when a test occurs the failure is detected and repaired instantaneously. The availability of such periodically tested blocks is equal to 1 just after a test and then decreases until the next test is performed where it comes back to 1 again. This leads to the typical "saw tooth" curves shown in Figure C.5 where all blocks are tested with the same test interval.



**Figure C.5 – Typical availability of RBD with periodically tested failures**

Safety systems implementing periodically tested components are easily modelled in this way. This is in particular the typical case of the safety instrumented systems described in the functional safety standards IEC 61508, IEC 61511 and ISO/TR 12489.

This combination of individual Markov processes through logical combinations has proven very useful for both RBD and FT approaches.

### C.5 Average and asymptotic (steady state) availability calculations

It is easy to calculate average availability  $A_S^{avg}(t_1, t_2)$  by a simple numerical averaging of the curve  $A_S(t)$ , as shown as a dotted line in Figure C.1 over a period of time  $[t_1, t_2]$ .



This calculation is valid in any case but when an asymptotic value  $A_S^{as} = \lim_{t \rightarrow \infty} A_S(t)$  is reached like in Figure C.5, this asymptotic value also gives the average availability as soon as  $t$  is long enough for allowing all the block availabilities to have reached the asymptotic values. Therefore, when the system availability has an asymptotic value, this is also the long term average availability:  $A_S^{avg} = A_S^{as}$ .

When the blocks are very quickly repaired ( $MTTR_i \ll MTTF_i$ ), the asymptotic values are reached very quickly (after a duration equal to 2 or 3 times the largest  $MTTR_i$ ) and this case is almost the same as the one with constant probabilistic values.

When as in Figure C.5 there is no asymptotic value, the average availability has to be calculated from  $A_S(t)$ . Nevertheless, in case of recurrent phases, the availability  $A_S(t)$  converges toward a recurrent profile after some phases. For example, for a simple block and when the repairs are not instantaneous, the recurrent profile is reached after 2 or 3 test intervals and the average availability during an interval converges to a limit value:

$$A_S^{lim}(\tau) = \lim_{n \rightarrow \infty} A_S^{avg}[n\tau, (n+1)\tau].$$

Models like Figure C.5 can be used to model safety systems and calculate the  $PFD_{avg}$  (average of the probability of failure on demand) required by the functional safety standards IEC 61508 and IEC 61511 for safety instrumented systems operating in low demand mode:

$$PFD_{avg} = U_S^{avg}(0, T) = 1 - A_S^{avg}(0, T).$$

When a recurrent profile exists, and this is often the case, then  $PFD_{avg} = U_S^{lim}(\rho) = 1 - A_S^{lim}(\rho)$  where  $\rho$  is the length of the recurring interval (see 10.3.2).

## C.6 Frequency calculations

In addition of the classical availability  $A_S(T)$  and reliability  $R_S(T)$ , the average failure frequency  $w_S^{avg}(0, T)$  is another relevant probabilistic indicator useful to characterize a system.

This parameter which does not exist for constant failure probabilities and is not useful for non-repaired systems is very useful when dealing with repaired systems which may fail (and be repaired) several times over a given period  $[0, T]$ . In this case, if  $n$  is the number of failures over this given time interval, the average failure frequency is given by  $n/T = w_S^{avg}(0, T)$ .

The average failure frequency can be calculated by using RBDs but the mathematics which implies the calculation of the Birnbaum importance factors (see Clause D.3) are not as simple as for availability and reliability calculations. Therefore, it is difficult to perform the calculations by hand but powerful algorithms are available to do that.

The calculation of the average frequency is performed in several steps.

- 1) Calculation of Birnbaum importance factors  $MIF_S(B_i, t)$  related to each block  $B_i$ . This importance factor is also called "marginal importance factor" (see Annex D).  $MIF_S(B_i, t)$  is obtained from the conditional availabilities  $A_{S|B_i}(t)$  and  $A_{S|\bar{B}_i}(t)$  by the following formula (see reference [14]):

$$MIF_S(B_i, t) = \frac{\partial[A_S(t)]}{\partial[A_{B_i}(t)]} = A_{S|B_i}(t) - A_{S|\bar{B}_i}(t) \quad (C.4)$$

- 2) Calculation of the unconditional failure intensities  $w_i(t)$  of each block  $B_i$ . This is obtained by combining the failure rate  $\lambda_i(t)$  and the availability  $A_{B_i}(t)$  of the related block:

$$w_i(t) = \lambda_i(t) \cdot A_{B_i}(t) \quad (C.5)$$

- 3) Calculation of the unconditional failure intensity of the system:

$$w_S(t) = \sum_i MIF_S(B_i, t) \cdot w_i(t) \quad (C.6)$$

- 4) Calculation of the expected number of failures  $W_S(T)$  over  $[0, T]$ . As the unconditional failure intensity  $w_S(t)$  is also the failure frequency (see 3.31) of the system at time  $t$ , the expected number of failures can be obtained by a simple integration:

$$W_S(T) = \int_0^T w_S(t) \cdot dt \quad (C.7)$$

- 5) Calculation of the average failure frequency:

$$w_S^{\text{avg}}(T) = \frac{1}{T} \int_0^T w_S(t) \cdot dt = \frac{W_S(T)}{T} \quad (C.8)$$

Except in very simple cases, Formula (C.8) cannot be done by hand and must be done numerically.

## C.7 Reliability calculations

While the average failure frequency  $w_S^{\text{avg}}(0, t)$  can be calculated in any case, the system reliability  $R_S(t)$  can be obtained by analytical calculations only in very particular cases:

- a) the RBD comprises only non-repaired blocks and in this case  $R_S(t) = A_S(t)$ ;
- b) the conditional failure intensity  $A_{vS}(t)$  reaches an asymptotic value  $A_{vS}^{\text{as}}$ .

In the case b) the conditional failure intensity  $A_{vS}(t)$  can be obtained from the unconditional failure intensity and the system availability by the following formula:

$$A_{vS}(t) = w_S(t) / A_S(t) \quad (C.9)$$

$A_{vS}(t)$  is also called Vesely failure rate as it can be used as an approximation of the system failure rate  $A_S(t)$  to perform reliability calculation from the classical equation:

$$R_S(t) \approx \exp\left(-\int_0^t A_{vS}(u) \cdot du\right) \quad (C.10)$$

Of course this is particularly useful when the system reaches a steady state because in this case  $A_{vS}(t)$  reaches a constant asymptotic value  $A_{vS}(t) \xrightarrow{t \rightarrow \infty} A_{vS}^{as}$ . This is typically the case of RBDs like those presented in Figure C.3 where the availability  $A_S(t)$ , the unconditional failure intensity  $w_S(t)$  and the conditional failure intensity  $A_{vS}(t)$  reach asymptotic values  $A_S^{as}$ ,  $w_S^{as}$ ,  $A_{vS}^{as}$ .

Then in this case the system failure rate  $\lambda_S$  can be approximated by  $\lambda_S \approx A_{vS}^{as} = w_S^{as} / A_S^{as}$  and the system reliability is obtained as:

$$R_S(t) \approx \exp(-A_{vS}^{as} \cdot t) \quad (\text{C.11})$$

The accuracy of the approximation given by Formula (C.11) is very good when the transient period has elapsed. This transient period is very short when the failures of the blocks of the RBD are quickly detected and repaired: i.e. Formula (C.11) can be used after two or three times the largest MTTR of the blocks.

All those calculations are not easy to perform by hand but the fast algorithms now available and based on BDD are able to process large RBDs for reliability calculation purposes.

For cases other than a) and b), other techniques, like Monte Carlo simulation (e.g. by using DRBDs, see 12.2 and Annex E), Markov or Petri net techniques, should be used instead.

## Annex D (informative)

### Importance factors

#### D.1 General

When analysing a system, it is useful to rank the components according to their impact on the probability of success (or of failure) of the system of interest. This can be done by using one or several of the importance factors which have been developed for this purpose (see references [12], [13], [14], [29] and [30]).

The following Clauses D.2 to D.9 describe the main importance factors and explain how to compute them when coherent RBDs are implemented. For the sake of simplicity, this is developed for the constant probability case but when the probabilities are time-dependent (e.g., repaired systems), the formulae are similar for a given value of the time  $t$ .

#### D.2 Vesely-Fussell importance factor

The Vesely-Fussell importance factor,  $FV_S(B_i)$ , is one of the most popular importance factors. It is based on the minimal cut sets of the system. It measures the probability that, when system S fails, the failure of the block  $B_i$  participates in at least one of the minimal cut sets having caused the failure of S. This importance factor takes into account both the probability of failure of  $B_i$  and the order of the minimal cut set that it belongs to. This is a rather accurate importance factor for measuring the impact of a component on the probability of failure of the system.

Let us consider  $C(\bar{b}_i)_j$  a minimal cut set containing  $\bar{b}_i$  (i.e. the failure of component  $B_i$ ). Then the Vesely-Fussell importance factor is given by:

$$FV_S(B_i, t) = \frac{P[\bigcup_j C(\bar{b}_i)_j]}{P_S} \quad (D.1)$$

Formula (D.1) is not very easy to calculate and, when the probability of system failure is low ( $P_S \ll 1$ ), the following approximation is often used instead:

$$FV_S(B_i, t) \approx \frac{\sum_j P[C(\bar{b}_i)_j]}{\sum_{i,j} P[C(b_i)_j]} \quad (D.2)$$

This Formula (D.2) is very easy to use by hand when the number of minimal cut sets is not too high: this is the sum of the probabilities of minimal cut sets containing the failure of  $B_i$  divided by the sum of all the minimal cut sets.

#### D.3 Birnbaum importance factor or marginal importance factor

The marginal importance factor  $MIF_S(B_i)$  is also called Birnbaum importance factor. It provides the basis for estimating the equivalent failure rate (and therefore the reliability) of a repaired system (see 10.3.1.4). It is based on the partial derivative of the probability of

success (or failure) of the system with regards to the probability of success (or failure) of the considered block  $B_i$ . The Birnbaum importance factor is basically given by the following Formula (D.3):

$$MIF_S(B_i) = \frac{\partial P_s}{\partial P_{b_i}} = \frac{\partial P_s^-}{\partial P_{b_i}^-} \quad (D.3)$$

It is symmetrical with regards to success or failure. It may be interpreted as the probability that the system is in a critical state (working or failed) due to the state of  $B_i$  i.e., if S is working the failure of  $B_i$  will fail S and if S is failed then the repair of  $B_i$  will cause the repair of S.

Formula (D.3) is equivalent to Formula (D.4):

$$MIF_S(B_i) = P_{s|b_i} - P_{s|b_i}^- = P_{s|b_i}^- - P_{s|b_i} \quad (D.4)$$

Therefore, this importance factor can be calculated by using the BDD calculations described in 11.6 for the conditional probabilities  $P_{s|b_i}$  and  $P_{s|b_i}^-$ .

It has to be noted that the Birnbaum importance factor does not depend on the probability of success (or failure) of the component  $B_i$ .

#### D.4 Lambert importance factor or critical importance factor

The critical importance factor  $CIF_S(B_i)$  is also called Lambert importance factor. This is a normalized Birnbaum importance factor. It is given by the following Formula (D.5):

$$CIF_S(B_i) = \frac{P_{b_i}^-}{P_s^-} MIF_S(B_i) = \frac{1 - P_{b_i}}{1 - P_s} MIF_S(B_i) \quad (D.5)$$

This importance factor is easy to calculate when  $MIF_S(B_i)$  has been calculated.

#### D.5 Diagnostic importance factor

The diagnostic importance factor  $DIF_S(B_i)$  is given by the conditional probability that  $B_i$  is failed given that S is failed. Therefore, it allows to determine which components have to be examined in priority when S is failed in order to repair it as soon as possible.

It is given by the following Formula (D.6):

$$DIF_S(B_i) = P_{b_i|s}^- \quad (D.6)$$

The following equivalent Formula (D.7) is easier to calculate:

$$DIF_S(B_i) = P_{b_i}^- \frac{P_{s|b_i}^-}{P_s^-} \quad (D.7)$$

This importance factor is linked to  $RAW_S(B_i)$  (see Clause D.6) by:

$$DIF_S(B_i) = P_{b_i}^- \cdot RAW_S(B_i) = (1 - P_{b_i}) \cdot RAW_S(B_i)$$

NOTE 1 When  $DIF_S(B_i)$  is low, the probability that  $B_i$  is failed when S is failed is low. When  $DIF_S(B_i)$  is high, the probability that  $B_i$  is failed when S is failed is also high. Therefore, the most useful is to examine  $B_i$  with intermediate values of  $DIF_S(B_i)$  to diagnose if they are failed or not.

NOTE 2 The repair of a failed component identified by using the DIF does not necessarily repair the system S.

## D.6 Risk achievement worth

The risk achievement worth  $RAW_S(B_i)$  is the conditional probability that S is failed given  $B_i$  is failed, normalized by the probability of failure of S. It allows to measure the increase of the probability of failure when  $B_i$  actually fails.

It is given by the following Formula (D.8):

$$RAW_S(B_i) = \frac{P_{s|b_i}^-}{P_s^-} = \frac{1 - P_{s|b_i}^-}{1 - P_s^-} \quad (D.8)$$

## D.7 Risk reduction worth

The risk reduction worth  $RRW_S(B_i)$  is the conditional probability that S is failed given  $B_i$  is not failed, normalized by the probability of failure of S. It allows to measure the reduction of the probability of failure when  $B_i$  actually works.

It is given by the following Formula (D.9):

$$RRW_S(B_i) = \frac{P_{s|b_i}}{P_s} = \frac{1 - P_{s|b_i}^-}{1 - P_s^-} \quad (D.9)$$

## D.8 Differential importance measure

The differential importance measure  $DIM_S(B_i)$  is a local sensitivity measure of  $P_{b_i}$  on  $P_s$  defined as follows:

$$DIM_S(B_i) = \frac{\frac{\partial P_s}{\partial P_{b_i}} dP_{b_i}}{\sum_{k=1}^n \frac{\partial P_s}{\partial P_{b_k}} dP_{b_k}} \quad (D.10)$$

In the Formula (D.10),  $\frac{\partial P_s}{\partial P_{b_i}} dP_{b_i}$  is the change of  $P_s$  induced by a small change ( $dP_{b_i}$ ) in the probability of block  $B_i$ .

The differential importance measure has two important properties:

- this is an additive measure:  $DIM_S(B_i, B_j) = DIM_S(B_i) + DIM_S(B_j)$ ;
- the sum of the differential importance measure of all blocks within an RBD equals unity:  $DIM_S(B_1, B_2, \dots, B_n) = DIM_S(B_1) + DIM_S(B_2) + \dots + DIM_S(B_n) = 1$ .

The differential importance measure  $DIM_S(B_i)$  is linked to other importance factors in special cases:

a) uniform change (criterion H1):  $\Delta P_{b_i} = \Delta P_{b_k} \ll 1$  for  $j, k = 1, 2, \dots, n$

$$DIM_S^{H1}(B_i) = \frac{MIF_S(B_i)}{\sum_{k=1}^n MIF_S(B_k)}$$

b) proportional relative changes (criterion H2):  $\frac{\Delta P_{b_i}}{P_{b_i}} = \frac{\Delta P_{b_k}}{P_{b_k}}, \Delta P_{b_i} = \Delta P_{b_k} \ll 1$  for  $i, k = 1, 2, \dots, n$

$$DIM_S^{H2}(B_i) = \frac{CIF_S(B_i)}{\sum_{k=1}^n CIF_S(B_k)}$$

The calculations of marginal importance factor are explained in Clause D.3 and that of critical importance factor in Clause D.4.

## D.9 Remarks about importance factors

There are many importance factors which have been developed for specific uses. It can be demonstrated that:

$$RAW_S(B_i) \geq DIF_S(B_i) \geq FV_S(B_i) \geq CIF_S(B_i)$$

Among them only the Vesely-Fussell importance factor can be handled by hand (when  $1 - P_s \ll 1$  and the number of minimal cut sets is not too high). The others imply the use of conditional probabilities difficult to handle by hand but easy to calculate by for example, using the BDD method described in 11.6.

Other importance factors have been developed to deal with non-coherent RBDs (see 12.2). They shall be used in this case as the importance factors described above are not valid and may lead to inconsistent results.

## Annex E (informative)

### RBD driven Petri nets

#### E.1 General

One effective way to deal with the dynamic RBDs is to mix the RBD and Petri net approaches. This allows to build large PNs and to use Monte Carlo simulation to calculate the probabilistic results of interest.

The simplest method is to model the blocks and the external elements by individual sub-PNs which interact through the use of predicates and assertions. Such model is an RBD driven Petri net (see reference [18]) which

- keeps the logical RBD structure for the logical calculation of the system state from the block states,
- takes advantage of the powerfulness of Petri nets to model the interactions between blocks and/or external elements.

#### E.2 Example of sub-PN to be used within RBD driven PN models

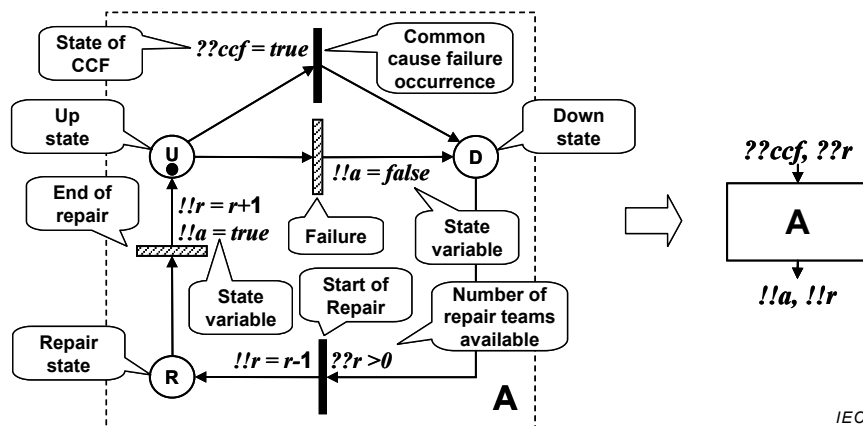


Figure E.1 – Example of a sub-PN modelling a DRBD block

Figure E.1 gives an example of a sub-PN developed to be used within a DRBD. The block is characterized by:

- three states: up (U), down (D) and repair (R);
- four transitions: (independent) failure, failure due to common cause failure, start of repair and end of repair;
- several predicates and assertions:
  - two assertions,  $!!a=true$  and  $!!a=false$ , to update the state of the block (up or down). Each block state is modelled in this way in order to evaluate the state of the whole system through the logical architecture of the RBD;
  - one predicate,  $??ccf = true$ , which triggers the failure of the block when the CCF occurs. This is used to model the interactions with an external element modelling the CCF;
  - one predicate,  $??r > 0$ , allowing to start the repair when at least one repair team is available. This is used to model the interactions between the blocks sharing the same repair teams;



- one assertion,  $!!r = r-1$ , to decrease the number of repair teams available by one when a repair is started. This is used to model the interactions between the blocks sharing the same repair teams;
- one assertion,  $!!r = r+1$ , to increase the number of repair teams available by one when a repair is completed. This is used to model the interactions between the blocks sharing the same repair teams.

On the right hand side of Figure E.1 is proposed a representation of the block related to this sub-PN.

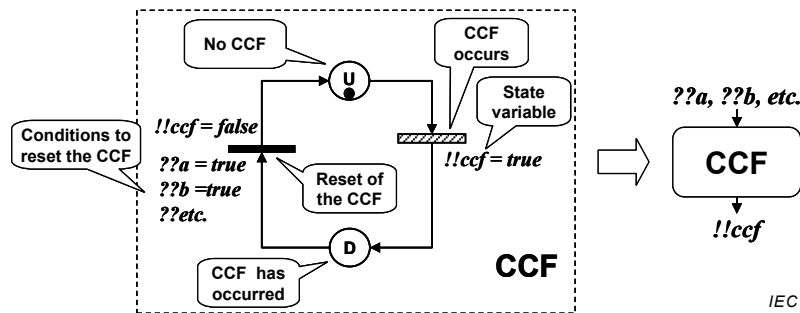


Figure E.2 – Example of a sub-PN modelling a common cause failure

Figure E.2 gives an example of a sub-PN developed to be used as an external element of a DRBD. It models a common cause failure characterized by

- two states: U (up: not occurred CCF), D (down: occurred CCF),
- two transitions: occurrence of CCF, reset of the CCF,
- several predicates and assertions:
  - two assertions,  $!!ccf=true$  and  $!!ccf=false$ , to update the state of the CCF (not-occurred or occurred). This is used to fail the blocks related to this CCF;
  - several predicates,  $??a = true$ ,  $??b = true$ , etc. which allow to reset the CCF only after all the blocks affected by this CCF have been repaired.

On the right hand side of Figure E.2 is proposed a representation of the external element related to this sub-PN.

Those sub-PNs may be used to build DRBD as this is shown in Figure E.3.

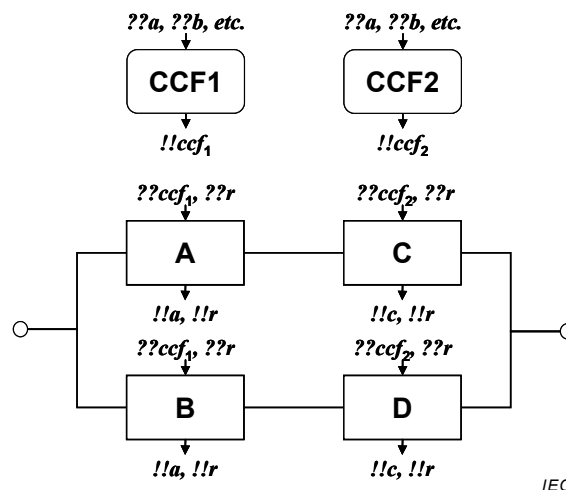


Figure E.3 – Example of DRBD based on RBD driven PN

This DRDB models

- a common cause failure on blocks A and B and another common cause failure on blocks C and D,
- a limited number of repair teams for repairing the four blocks. The number of repair teams is given by the initial conditions:  $r = 1$  models a single repair team,  $r = 2$  models two available repair teams, etc.,  $r = 4$  is equivalent to the classical assumption considering that there are as many repair teams as repaired blocks.

### E.3 Evaluation of the DRBD state

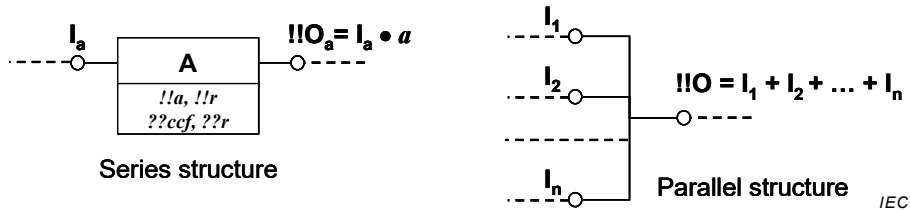


Figure E.4 – Logical calculation of classical RBD structures

The state of the system is given by the combinations of the states of the blocks ( $a, b, c$  and  $d$ ) and this can be done exactly as for an ordinary RBD by using the global assertions presented in Figure E.4:

- $!!O_a = I_a \cdot a$  for series structures: the output of block A is up if its input is up and if the block is in up state;
- $!!O = I_1 + I_2 + \dots + I_n$  for parallel structures: the output is up if at least one of the inputs is up.

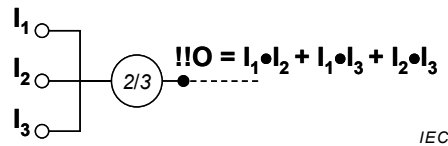
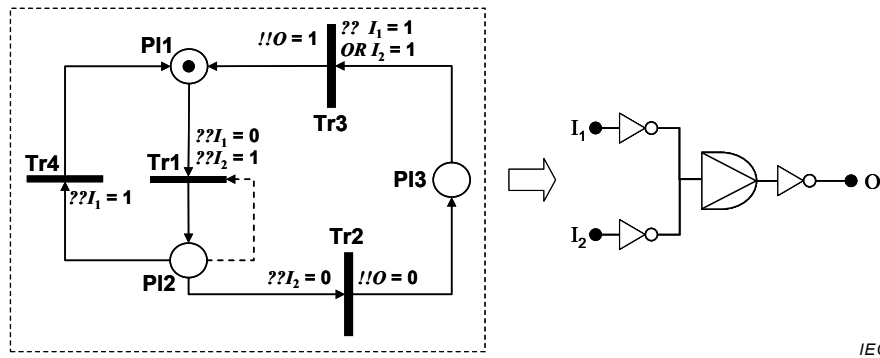


Figure E.5 – Example of logical calculation for an  $n/m$  gate

Figure E.5 shows that a  $2/3$  gate can also be calculated by a simple logical formula. This can be easily extended to any kind of  $n/m$  gates.

For the PAND gate, no logical formula exists but a simple sub-PN such as the one presented in Figure E.6 can be used instead. This PN has been drawn for two inputs but can be easily extended to  $n$  inputs. It is equivalent to the finite-state automaton presented in Figure 50.



IEC

**Figure E.6 – Example of sub-PN modelling a PAND gate with 2 inputs**

The behaviour of this PN is the following:

- 1) at the beginning, the output is in the up state ( $O = 1$ ) and the place PI1 is marked with 1 token;
- 2) if  $I_2$  is false ( $I_2 = 0$ ) when  $I_1$  is true ( $I_1 = 1$ ) the transition Tr1 is inhibited;
- 3) if  $I_1$  becomes false ( $I_1 = 0$ ) when  $I_2$  is true ( $I_2 = 1$ ) that means that  $I_1$  has occurred before  $I_2$  and then the transition Tr1 is immediately fired. The token is removed from PI1 and one token is added in PI2. This inhibits Tr1 (thanks to the inhibitor arrow in dotted lines) and validates Tr2 and Tr4;
- 4) if  $I_1$  is true ( $I_1 = 1$ ) before  $I_2$  becomes false, then Tr4 is fired and the PN comes back to its initial state;
- 5) if  $I_2$  becomes false ( $I_2 = 0$ ) while  $I_1$  is still false ( $I_1 = 0$ ), the transition Tr2 is immediately fired and the output becomes false ( $O = 0$ );
- 6) if  $I_1$  or  $I_2$  become true again ( $I_1 = 1$  or  $I_2 = 1$ ), then Tr3 is fired and the output becomes true again ( $O = 1$ );
- 7) if Tr3 has been fired because  $I_1 = 1$  the PN comes back to step 2 where Tr1 is inhibited;
- 8) if Tr3 has been fired because  $I_2 = 1$  the PN comes back to step 3 and Tr1 is immediately fired.

With this sub-PN the output becomes "false" ( $O = 0$ ) only if  $I_1$  and  $I_2$  become "false" ( $I_1 = 0$ ,  $I_2 = 0$ ) and in this order.

The same sub-PN can be used to model the finite-state automaton presented in Figure 52 for a SEQ gate but it is not sufficient to model the dynamic interaction between  $I_2$  and  $I_1$ :  $I_2$  cannot go into the down state before  $I_1$  has gone to the down state. This can be achieved, for example, by modelling blocks C and D in Figure 51 by sub-PNs like the one presented in Figure E.7 for block C.

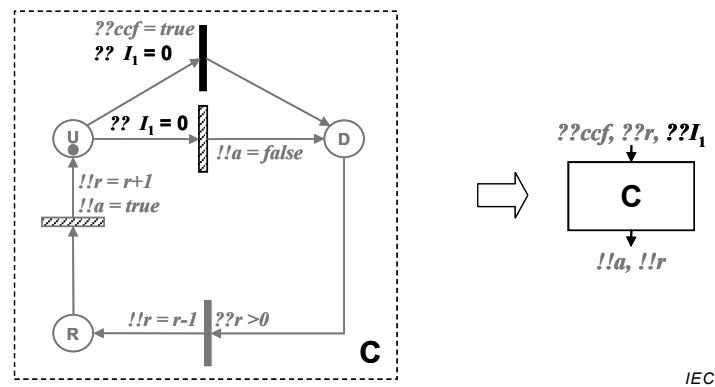


Figure E.7 – Example of the inhibition of the failure of a block

This sub-PN is derived from the one presented in Figure E.1 where the failure transitions of the block (independent and common cause failures) are inhibited as long as  $I_1$  has not gone to the down state ( $I_1 = 0$ ).

#### E.4 Availability, reliability, frequency and MTTF calculations

When the model is built, it can be used for probabilistic calculation and this is achieved by using Monte Carlo simulation. The sub-PN presented in Figure E.8 models the DRBD output and it can be used to obtain all the needed probabilistic results:

- the marking of place U at time  $t$  gives the system availability  $A_S(t)$ ;
- the marking of place D at time  $t$  gives the system unavailability  $U_S(t)$ ;
- the mean marking of place U at over  $[0, T]$  gives the average system availability  $A(0, T)$  over  $[0, T]$ ;
- the mean marking of place D at over  $[0, T]$  gives the average system unavailability  $U(0, T)$  over  $[0, T]$ ;
- the frequency of firing of the transition "First failure" gives the system unreliability  $R_S(t)$  over  $[0, t]$ ;
- the frequency of firing of the transition "failure" gives the average system failure frequency  $w_S^{\text{avg}}(0, t)$ ;
- the mean marking of the place M gives the mean time before the first failure occurs. When the time is long enough to have at least one failure per simulation, then this gives the MTTF of the system modelled by the DRBD;
- etc.

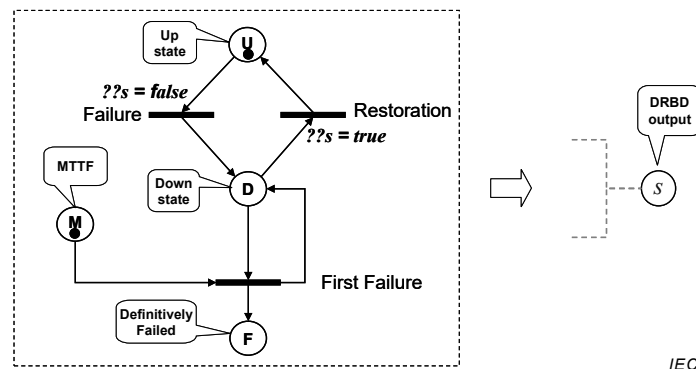


Figure E.8 – Sub-PN for availability, reliability and frequency calculations

## Annex F (informative)

### Numerical examples and curves

#### F.1 General

Annex F develops some numerical examples on a typical RBD structure and establishes the corresponding availability (definition 3.21), reliability (definition 3.26), conditional failure intensity (Vesely failure rate) (definition 3.30) and unconditional failure intensity (failure frequency) (definition 3.31). Curves are drawn in order to show how those parameters vary when time elapses.

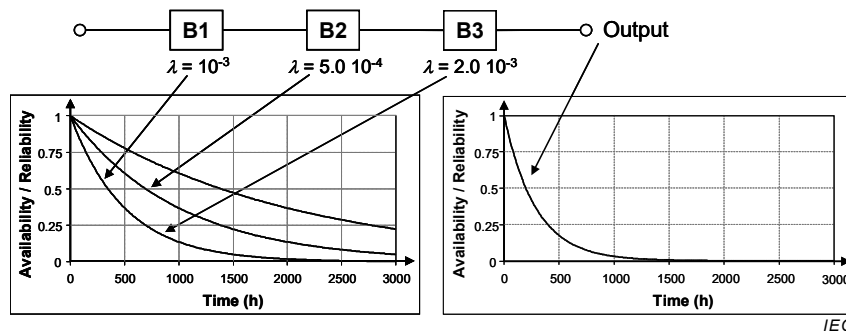
Analytical calculations are performed by using a tool implementing the BDD approach in order to propose results without approximations.

In Clause F.5 the Monte Carlo simulation approach is used to calculate the availability of dynamic RBDs involving several functional dependencies.

#### F.2 Typical series RBD structure

##### F.2.1 Non-repaired blocks

Figure F.1 represents a typical RBD series structure made of 3 non-repaired blocks. In this case the reliability and availability of the blocks are equal and this is the same for the whole system.



**Figure F.1 – Availability/reliability of a typical non-repaired series structure**

The left hand side of the figure represents the reliability/availability of the blocks which are modelled by constant failure rates (exponential laws):

- Block B<sub>1</sub>:  $\lambda_1 = 1,0 \times 10^{-3} \text{ h}^{-1}$ ,
- Block B<sub>2</sub>:  $\lambda_2 = 5,0 \times 10^{-4} \text{ h}^{-1}$ ,
- Block B<sub>3</sub>:  $\lambda_3 = 2,0 \times 10^{-3} \text{ h}^{-1}$

The right hand side of the figure represents the reliability/availability of the whole system.

Figure F.2 represents the failure rate,  $\Lambda(t)$ , and the failure frequency  $w(t)$  of the non-repaired series structure.

In this case the Vesely failure rate  $\Lambda_V(t)$  (conditional failure intensity) and the failure rate  $\Lambda(t)$  are equal and constant. As expected  $\Lambda(t) = \Lambda_V(t) = \lambda_1 + \lambda_2 + \lambda_3 = \text{constant}$ .

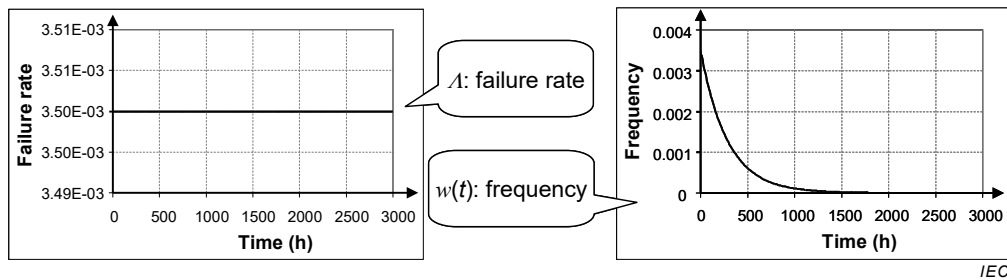


Figure F.2 – Failure rate and failure frequency related to Figure F.1

Therefore, the system made of three blocks is equivalent to a single block C with  $\lambda_C = \lambda_1 + \lambda_2 + \lambda_3$ .

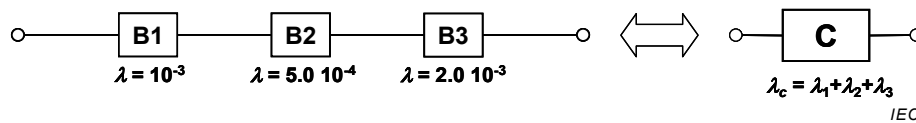


Figure F.3 – Equivalence of a non-repaired series structure to a single block

The failure frequency (unconditional failure intensity) decreases as the time  $t$  increases and goes to 0 when  $t$  goes to infinity. This is due to the fact that, being non-repaired, the system can fail only once.

## F.2.2 Repaired blocks

Figure F.4 represents a typical RBD series structure made of 3 repaired blocks. In this case the availability and the reliability of the blocks are different and this is the same for the whole system.

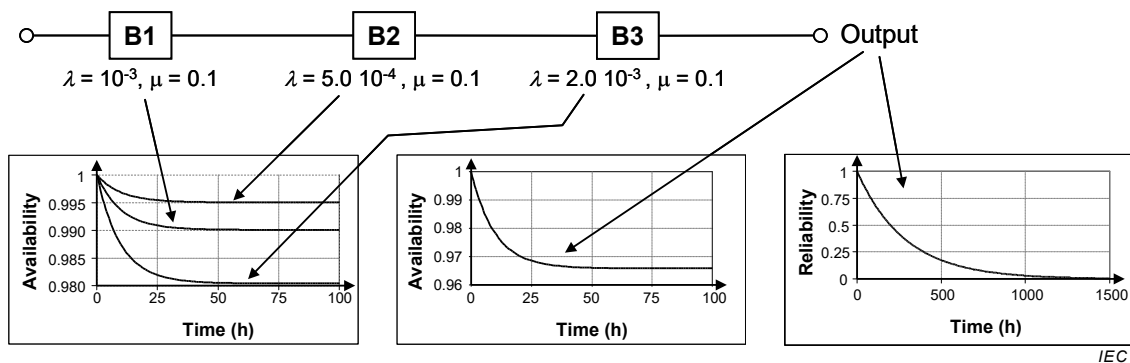


Figure F.4 – Availability/reliability of a typical repaired series structure

The left hand side of the figure represents the availability of the blocks which are modelled by the following constant failure and repair rates:

- Block B<sub>1</sub>:  $\lambda_1 = 1,0 \times 10^{-3} \text{ h}^{-1}$ ,  $\mu = 0,1 \text{ h}^{-1}$ ;
- Block B<sub>2</sub>:  $\lambda_2 = 5,0 \times 10^{-4} \text{ h}^{-1}$ ,  $\mu = 0,1 \text{ h}^{-1}$ ;
- Block B<sub>3</sub>:  $\lambda_3 = 2,0 \times 10^{-3} \text{ h}^{-1}$ ,  $\mu = 0,1 \text{ h}^{-1}$ .

The system availability is presented in the middle of the figure and the reliability on the right hand side.

The behaviour is very different compared to the non-repaired case: as shown in Figure F.4, the availabilities of the blocks (left hand side) as well as the availability of the whole system (middle of the figure) quickly reach asymptotic values.

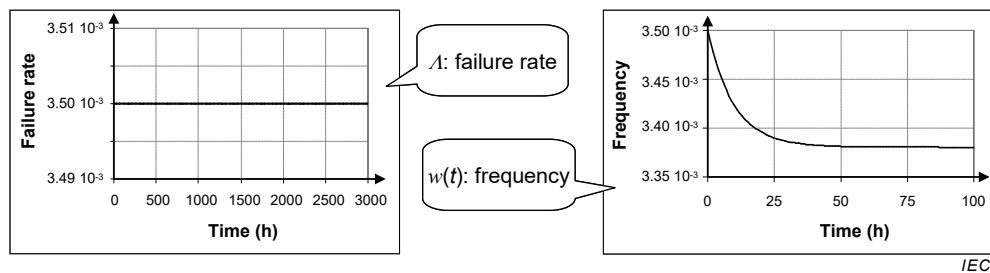


Figure F.5 – Failure rate and failure frequency related to Figure F.4

Figure F.5 represents the failure rate,  $\lambda(t)$ , and the failure frequency  $w(t)$  of the repaired series structure.  $\lambda(t)$  is the same as in non-repaired case because each block failure causes the whole system failure and then, from a reliability calculation point of view cannot be repaired (see 10.3.3).

### F.3 Typical parallel RBD structure

#### F.3.1 Non-repaired blocks

Figure F.6 represents a typical RBD parallel structure made of 3 non-repaired blocks. In this case the reliability and availability of the blocks are equal and this is the same for the whole system.

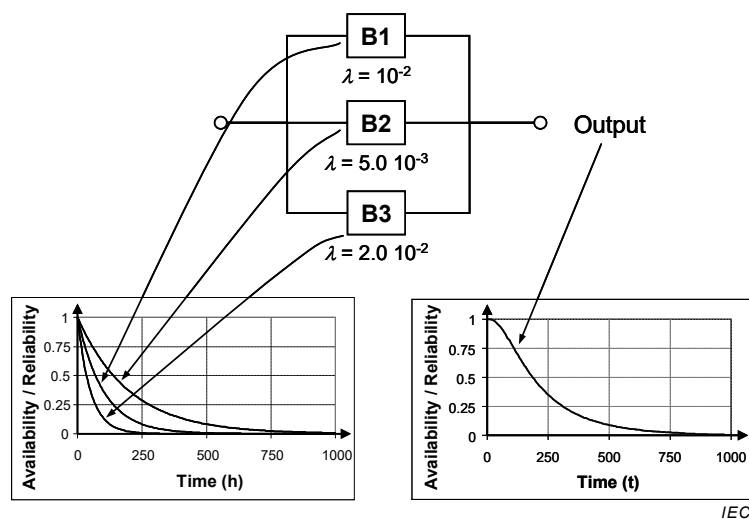


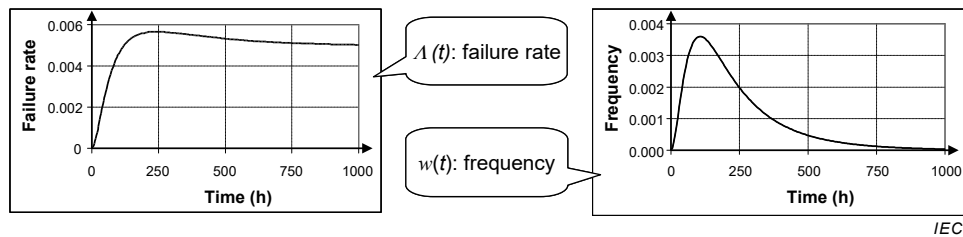
Figure F.6 – Availability/reliability of a typical non-repaired parallel structure

The left hand side of the figure represents the availability/reliability of the blocks which are modelled by constant failure rates (exponential laws):

- block B<sub>1</sub>:  $\lambda_1 = 1,0 \times 10^{-2} \text{ h}^{-1}$ ;
- block B<sub>2</sub>:  $\lambda_2 = 5,0 \times 10^{-3} \text{ h}^{-1}$ ;
- block B<sub>3</sub>:  $\lambda_3 = 2,0 \times 10^{-2} \text{ h}^{-1}$ .

The right hand side of the figure represents the reliability/availability of the whole system.

Figure F.7 represents the failure rate,  $\lambda(t)$ , and the failure frequency,  $w(t)$ , of the non-repaired parallel structure. As the availabilities and reliabilities are the same, the failure rate and the Vesely failure rate are also the same:  $\lambda(t) = \lambda_V(t)$ .



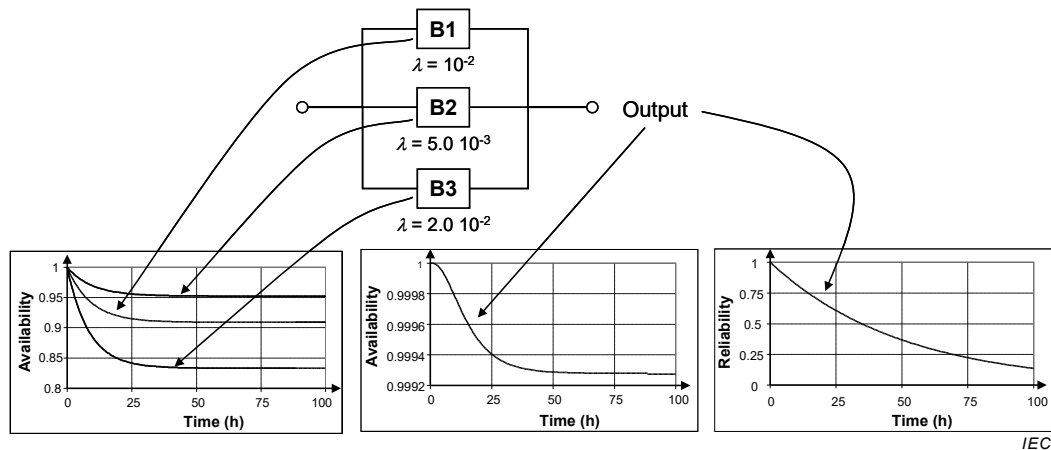
**Figure F.7 – Failure rate and failure frequency related to Figure F.6**

The behaviour is very different compared to the non-repaired series structure:

- $\lambda(t)$  needs a very long time to reach an asymptotic value which is equal to that of the lower failure rate of the three blocks. This asymptotic value is reached when the blocks with higher failure rates have had time to fail. It is reached very slowly and cannot be used as an approximation of the failure rate.
- the failure intensity,  $w(t)$ , goes through a maximum value before decreasing to zero.

### F.3.2 Repaired blocks

Figure F.8 represents a typical RBD parallel structure made of 3 repaired blocks. In this case the reliability and availability of the blocks are different and this is the same for the whole system.



**Figure F.8 – Availability/reliability of a typical repaired parallel structure**

The left hand side of the figure represents the availability of the blocks which are modelled by the following constant failure and repair rates:

- block B<sub>1</sub>:  $\lambda_1 = 1,0 \times 10^{-2} \text{ h}^{-1}$ ,  $\mu = 0,1 \text{ h}^{-1}$ ;
- block B<sub>2</sub>:  $\lambda_2 = 5,0 \times 10^{-3} \text{ h}^{-1}$ ,  $\mu = 0,1 \text{ h}^{-1}$ ;
- block B<sub>3</sub>:  $\lambda_3 = 2,0 \times 10^{-2} \text{ h}^{-1}$ ,  $\mu = 0,1 \text{ h}^{-1}$ .

The system availability is presented in the middle of the figure and the reliability on the right hand side.



The behaviour is very different compared to the non-repaired case: as shown in Figure F.8, the availabilities of the blocks (left hand side) as well as the availability of the whole system (middle of the figure) quickly reach asymptotic values.

Figure F.9 represents the Vesely failure rate (conditional failure intensity),  $\Lambda_V(t)$ , and the failure frequency  $w(t)$  of the repaired parallel structure.

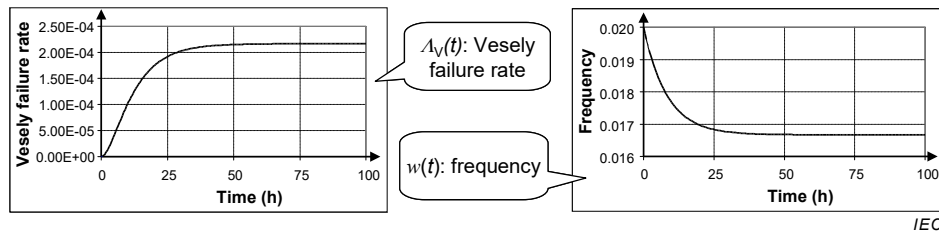


Figure F.9 – Vesely failure rate and failure frequency related to Figure F.8

The behaviour is very different compared to the non-repaired parallel structure:

- $\Lambda_V(t)$  very quickly reaches an asymptotic value. In this case, it becomes constant after 3 or 4 MTRR (30 h to 40 h) and this asymptotic value can be used as a constant failure rate to calculate the system reliability;
- the failure intensity,  $w(t)$ , also very quickly reaches an asymptotic value which can be used to calculate the average system failure frequency.

### F.4 Complex RBD structures

#### F.4.1 Non series-parallel RBD structure

Figure F.10 represents the RBD with a common block introduced in 7.5.2. This is a structure which cannot be reduced to simple series or parallel structures.

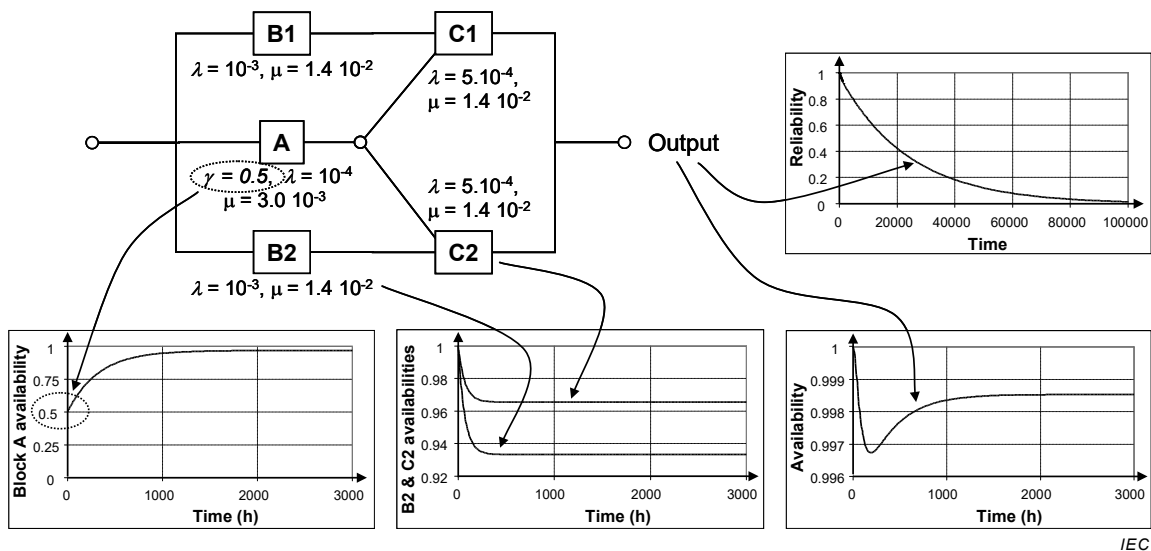


Figure F.10 – Example 1 from 7.5.2

The left hand side and the middle of the figure represent the availability of the blocks which are modelled by the following constant failure and repair rates:

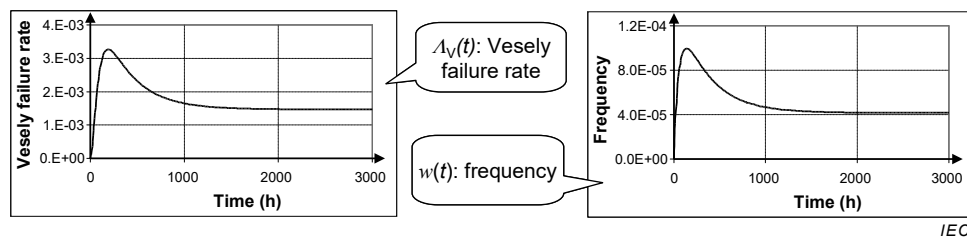
- blocks  $B_1$  and  $B_2$ :  $\lambda_1 = 1,0 \times 10^{-4} \text{ h}^{-1}$ ,  $\mu = 0,014 \text{ h}^{-1}$ ;
- blocks  $C_1$  and  $C_2$ :  $\lambda_2 = 5,0 \times 10^{-5} \text{ h}^{-1}$ ,  $\mu = 0,014 \text{ h}^{-1}$ ;

- block A:  $\lambda_3 = 1,0 \times 10^{-5} \text{ h}^{-1}$ ,  $\mu = 3,0 \times 10^{-3} \text{ h}^{-1}$ ,  $\gamma = 0,5$ .

The blocks are common repaired blocks but the repair rate of A is longer than the others and this block also has a probability of 0,5 to be in up state at time  $t$  equal to 0. As a result and as shown on Figure F.10, the availability of this block does not behave as the availability of the other blocks.

The system availability and the system reliability are presented on the right hand side of the figure. Due to the behaviour of A, the availability goes to a minimum before reaching an asymptotic value. This minimum corresponds to the MTTR of A.

The reliability behaves as usual.

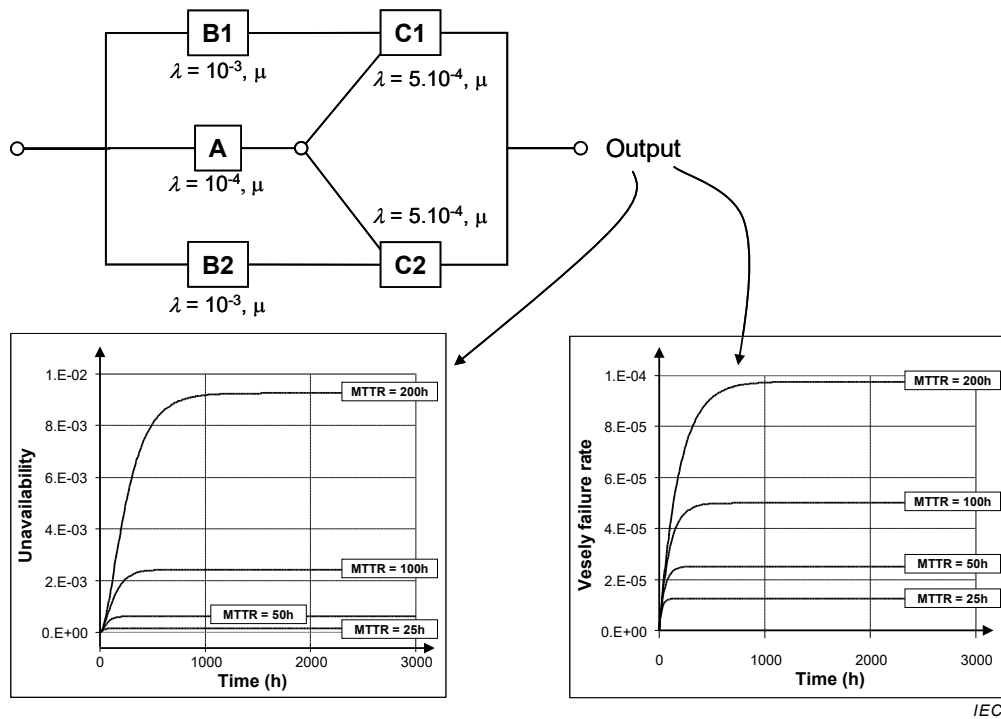


**Figure F.11 – Failure rate and failure frequency related to Figure F.10**

Figure F.11 represents the Vesely failure rate (conditional failure intensity),  $\lambda_V(t)$ , and the failure frequency  $w(t)$  of the system presented in Figure F.10. The shape of these parameters is due to the special behaviour of A:  $\lambda_V(t)$  as well as  $w(t)$  reach a maximum value before reaching asymptotic values

#### **F.4.2 Convergence to asymptotic values versus MTTR**

The unavailability and the equivalent failure rate of the RBD shown in Figure F.12 have been calculated with 4 different constant repair rates in order to visualize the impact of the MTTR on the speed of the convergence toward asymptotic values.



**Figure F.12 – Impact of the MTTR on the convergence quickness**

The blocks of the RBD presented in Figure F.12 have constant failure and repair rates. Therefore, they have markovian behaviours and this is why the system availability, system unavailability and conditional failure intensity (Vesely failure rate) converge toward asymptotic values.

Figure F.12 shows clearly that the convergence speed increases when the MTTR decreases. Then, when the system is quickly repaired, it behaves as if it had

- constant probabilities of success or of failures for the availability or unavailability calculations,
- constant failure rate for the reliability calculations.

#### **F.4.3 System with periodically tested components**

Figure F.13 represents the same RBD as above but with periodically tested components.

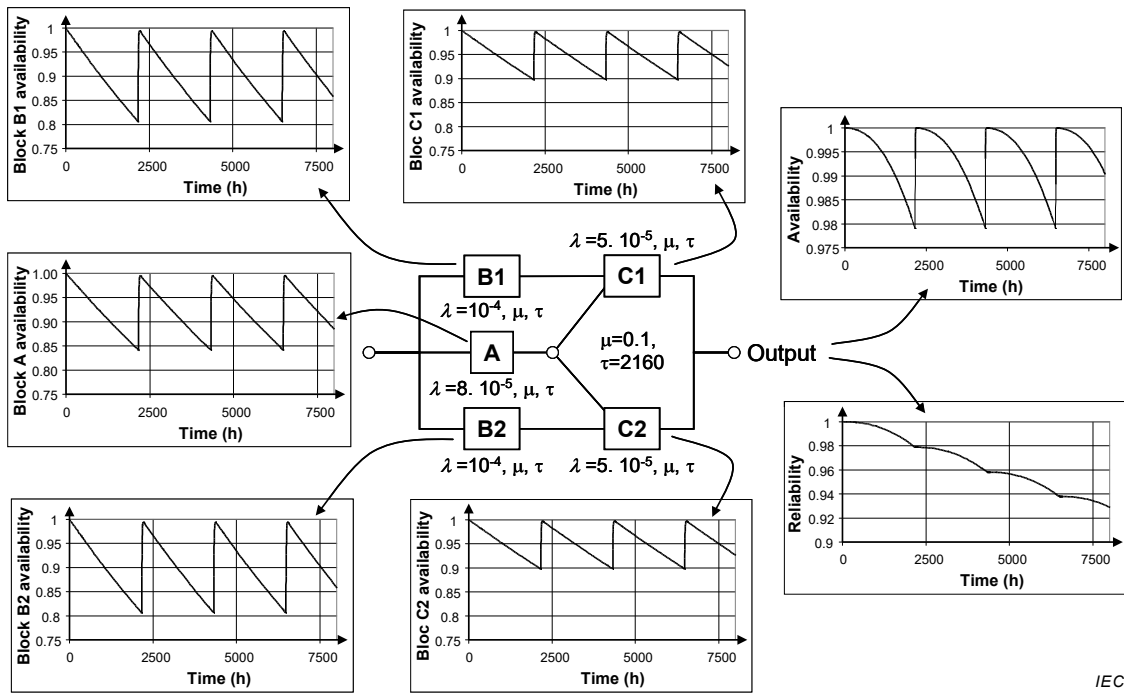


Figure F.13 – System with periodically tested blocks

The left hand side and the middle of the figure represents the availability of the blocks which are periodically tested and modelled by the following test intervals and constant failure and repair rates:

- blocks B<sub>1</sub> and B<sub>2</sub>:  $\lambda_1 = 1,0 \times 10^{-4} \text{ h}^{-1}$ ,  $\mu = 0,1 \text{ h}^{-1}$ ,  $\tau = 2\ 160 \text{ h}$ ;
- blocks C<sub>1</sub> and C<sub>2</sub>:  $\lambda_2 = 5,0 \times 10^{-5} \text{ h}^{-1}$ ,  $\mu = 0,1 \text{ h}^{-1}$ ,  $\tau = 2\ 160 \text{ h}$ ;
- block A:  $\lambda_3 = 8,0 \times 10^{-5} \text{ h}^{-1}$ ,  $\mu = 0,1 \text{ h}^{-1}$ ,  $\tau = 2\ 160 \text{ h}$ .

The availabilities of the blocks are typical saw tooth curves and this is the same for the system availability on the right hand side.

The test intervals give the special shape to the system reliability shown on the right hand side of Figure F.13. This is still a non-increasing function.

Such RBDs are commonly encountered when dealing with the functional safety of safety instrumented systems where some dangerous failures are detected by periodical tests. The average unavailability of these systems is called PFD<sub>avg</sub> (see 3.24 and IEC 61508 [5]).

Figure F.14 illustrates the Vesely failure rate and the failure frequency of such a system made of periodically tested components.

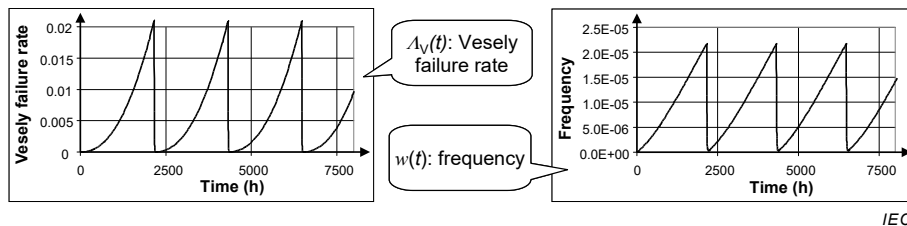
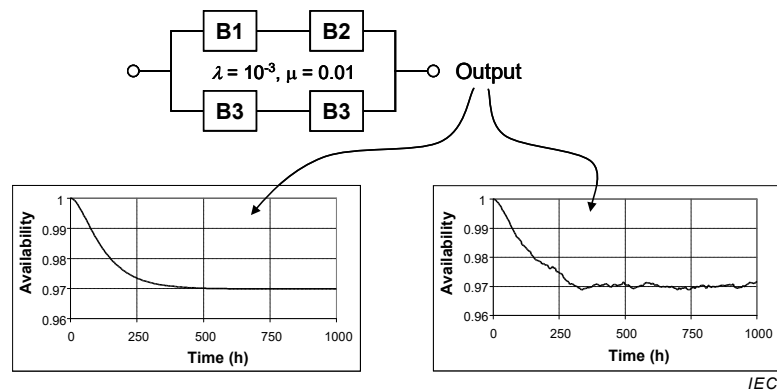


Figure F.14 – Failure rate and failure frequency related to Figure F.13

## F.5 Dynamic RBD example

### F.5.1 Comparison between analytical and Monte Carlo simulation results

Figure F.15 represents a small parallel-series RBD structure made of 4 similar blocks with the same failure and repair rate:  $\lambda = 1,0 \times 10^{-3} \text{ h}^{-1}$ ,  $\mu = 0,01 \text{ h}^{-1}$ .



**Figure F.15 – Analytical versus Monte Carlo simulation results**

The results obtained by a classical analytical calculation implementing the BDD approach are presented on the left hand side of Figure F.15 and those obtained by Monte Carlo simulation are presented on the right hand side.

The Monte Carlo results have been obtained in about 10 s on an ordinary laptop computer and 50 000 histories have been simulated. Of course the analytical curve is smoother than the one obtained from Monte Carlo simulation but the shape is the same and the two curves provide the same average availability value of 0,973 9 over 1 000 h and converge toward the same asymptotic value of 0,97.

### F.5.2 Dynamic RBD example

Several dynamic dependencies have been added to the previous RBD presented in F.5.1 in order to see which impact they can have on the results:

- common cause failures on B1 and B3, and common cause failures on B2 and B4 ( $\lambda_{CCF} = 1,0 \times 10^{-4}$ );
- single repair team;
- both a single repair team and common cause failures.

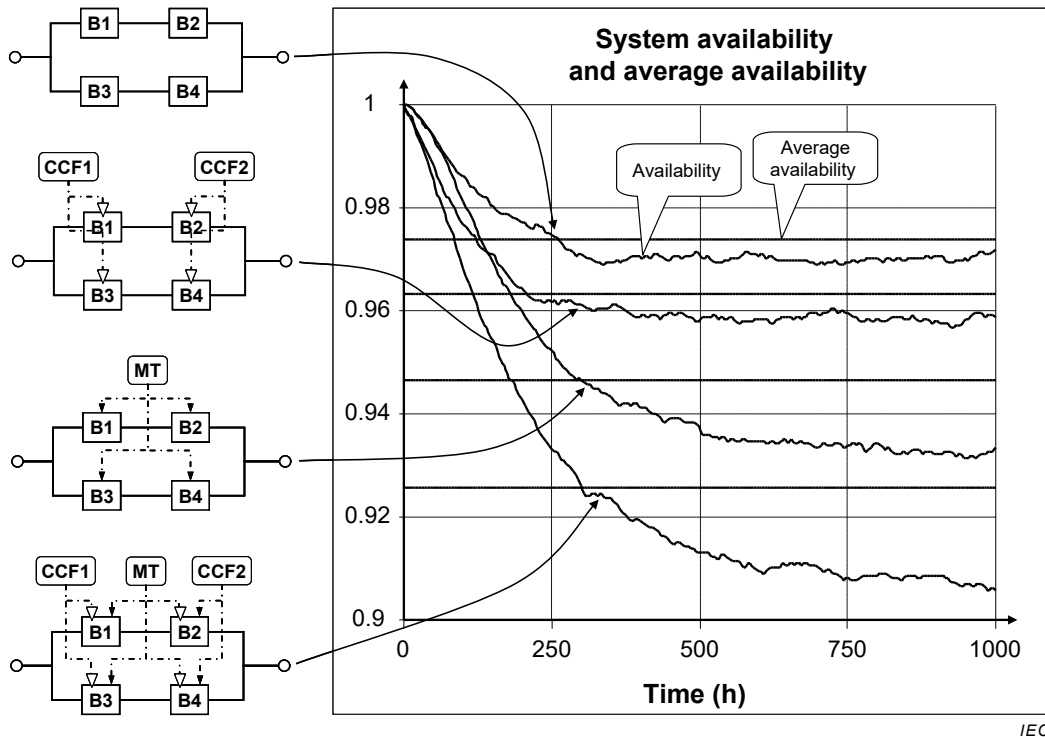


Figure F.16 – Impact of CCF and limited number of repair teams

Figure F.16 shows clearly that the impacts are not negligible and this is analysed in more details in Table F.1:

Table F.1 – Impact of functional dependencies

Configuration	System availability		System unavailability	
	$A_{avg}(1000)$	$A^{as}$	$U_{avg}(1000)$	$U^{as}$
No functional dependencies	$9,74 \times 10^{-1}$	$9,71 \times 10^{-1}$	$2,6 \times 10^{-2}$	$2,9 \times 10^{-2}$
CCFs	$9,63 \times 10^{-1}$	$9,59 \times 10^{-1}$	$3,7 \times 10^{-2}$	$4,1 \times 10^{-2}$
Single repair team	$9,47 \times 10^{-1}$	$9,32 \times 10^{-1}$	$5,4 \times 10^{-2}$	$6,8 \times 10^{-2}$
CCF + single repair team	$9,26 \times 10^{-1}$	$9,06 \times 10^{-1}$	$7,4 \times 10^{-2}$	$9,4 \times 10^{-2}$

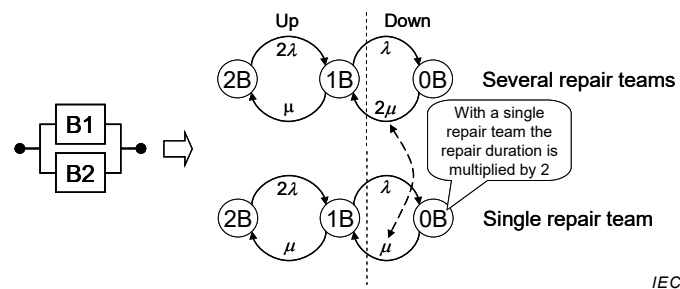
The impact is more visible when considering the unavailability rather than the availability. For example, for the asymptotic unavailabilities, the rates are:

- common cause failures: 140 %;
- single repair team: 234 %;
- both: 323 %.

Therefore, the assumption that there is as many repair teams as blocks is not really neutral and is non-conservative. The impact increases when:

- a) the block failure rates increase (the probability to have several failures at the same time increases),
- b) the MTTR increases (the probability to have the failure of one block during the repair of another one increases),
- c) the order of the preponderant minimal cut sets increases.

When the blocks are very reliable and when the MTTR is short, items a) and b) have a very limited impact. The main problem occurs with item c).



IEC

**Figure F.17 – Markov graphs modelling the impact of the number of repair teams**

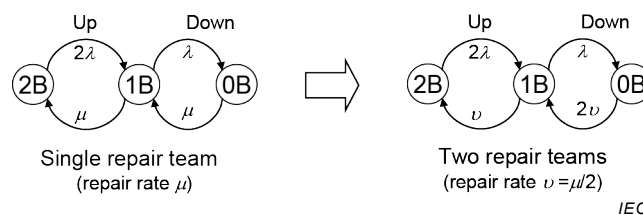
Let us imagine an RBD made of two similar blocks B1 and B2 with the same failure and repair rates ( $\lambda$ ,  $\mu$ ). Figure F.17 proposes the Markov graphs drawn in the case of two repair teams (top of the figure) and of a single repair team (bottom of the figure). Those Markov graphs have 3 states:

- 2B: 2 blocks in up states;
- 1B: 1 block in up state and 1 block in down state;
- 0B: 2 blocks in down states (0 blocks in up states).

The system is failed when both B1 and B2 are failed (state 0B). The sojourn time in this state is

- $1/\mu = \text{MTTR}$  when there is only a single repair team,
- $1/2\mu = 2 \times \text{MTTR}$  when there are several repair teams.

Then, with a single repair team, the mean time to repair the system is twice the needed mean time to repair when there are several teams. Therefore, when there is only a single repair team, a conservative approach should be to use an MTTR equal to twice that used with several repair teams. This is illustrated in Figure F.18 where the Markov graph on the right hand side (two repair teams with a repair rate  $\nu = \mu/2$ ) is an approximation of the Markov graph on the left hand side (one single repair team with a repair rate  $\mu$ ). Therefore the MTTR of each block has been multiplied by 2 on the right hand side.



IEC

**Figure F.18 – Approximation for two redundant blocks**

This approach is conservative because the mean sojourn time in the state 1B has been multiplied by 2. Nevertheless, even in this simple case, it may be too conservative. In addition, this may be difficult to apply for larger or more complex RBDs and it would be better to use, for example, an RBD driven PN (see Annex E) and perform Monte Carlo simulations which are now achievable on simple laptop computers.

## Bibliography

- [1] IEC 61025, *Fault tree analysis (FTA)*
- [2] IEC 61165, *Application of Markov techniques*
- [3] IEC 62551, *Analysis techniques for dependability – Petri net techniques*
- [4] IEC 60812, *Analysis techniques for system reliability – Procedure for failure mode and effects analysis (FMEA)*
- [5] IEC 61508:2010 (all parts), *Functional safety of electrical/electronic/programmable electronic safety-related systems*
- [6] IEC 61511:2016 (all parts), *Functional safety – Safety instrumented systems for the process industry sector*
- [7] ISO/TR 12489, *Petroleum, petrochemical and natural gas industries – Reliability modelling and calculation of safety systems*

### RBD methods (general)

- [8] Karnaugh, M., *The Map Method for Synthesis of Combinational Logic Circuits*. Trans. AIEE. pt. I, Vol. 72, No. 9., pp. 593-598 , 1953
- [9] Mc Cluskey Jr, E., *Minimization of Boolean functions*. Bell System Technical Journal, Vol. 35 , Issue 6, pp 1417 – 1444, 1956
- [10] Veitch, E. W., *A chart method for simplifying truth functions*. In Proceedings of the 1952 ACM national meeting, Pittsburgh, pp. 127-133). ACM, 1952
- [11] Umezawa, T., *A method of two-level simplification of Boolean functions*. Nagoya Mathematical Journal, Vol. 29, pp 201-210. 1967
- [12] Dutuit Y., Rauzy A. *Efficient algorithms to assess component and gate importance in fault tree analysis*, Reliability Engineering and System Safety, Vol. 72, No. 1, pp 213-222, 2001
- [13] Borgonovo E and Apostolakis G.E. *A New Importance Measure for Risk Informed Decision-Making*, Reliability Engineering and System Safety, Vol. 72 No. 2, pp 193-212, 2001
- [14] Borgonovo E. *Differential, Criticality and Birnbaum Importance Measures and Application to Basic Events, Groups and SSCs in Event Trees and Binary Decision Diagrams*, Reliability Engineering and System Safety, Vol. 92, No. 10, pp 1458-1467, 2007
- [15] Distefano, S., *System Dependability and performances: Techniques, Methodologies and Tools*. Thesis of doctor in philosophy, University of Messina, Italy, 2005
- [16] Bobbio, A., Codetta, D., *Parametric Fault trees with dynamic gates and repair boxes*, RAMS 2004, pp 459-465, 2004
- [17] Simeu-Abazi, Z. et als, *A methodology of alarm filtering using dynamic fault tree*, Reliability Engineering and System Safety, Vol. 96, No. 2, pp 257–266, 2011



- [18] Signoret, J.-P. & al, *Make your Petri nets understandable: Reliability block diagrams driven Petri nets*. Reliability Engineering and System Safety, Vol. 113, pp 61–75, 2013
- [19] Bennetts, R.G. *Analysis of Reliability Block Diagrams by Boolean Techniques*, IEEE Transactions on reliability, Vol. 31, No. 2, pp. 159-166, 1982
- [20] Barlow R.E., Proschan F., *Statistical Theory of Reliability and Life Testing. Probabilistic Models*, New York, Holt, Rinehart and Winston, 1975
- [21] Billinton R., Allan R.N., *Reliability Evaluation of Engineering Systems. Concepts and Techniques*. Second Edition, Springer, 1992
- [22] Birolini A., *Quality and Reliability of Technical Systems. Theory – Practice – Management*. Berlin, Springer Verlag, 1997
- [23] Gaede K.W., *Zuverlässigkeit, Mathematische Modelle*. München, Carl Hanser Verlag, 1977
- [24] Rausand M. and Hoyland A. *System Reliability Theory – Models, Statistical Methods and Applications*, second edition, Wiley, New York, 2004
- [25] Kaufmann A., Grouchko D., Cruon R., *Mathematical Models for the Study of the Reliability of Systems*, New York, Academic Press, 1977
- [26] Kuo W., Zuo M.J., *Optimal Reliability Modeling: Principles and Applications*. New York, Wiley, 2003
- [27] Lewis E.E., *Introduction to Reliability Engineering*, Second Edition, Wiley, New York, 1996
- [28] MIL-HDBK-338B, *Military Handbook Electronic Reliability Design Handbook*, 1 October 1998
- [29] Pagès A., Gondran A., *System Reliability. Evaluation and Prediction in Engineering*, Berlin, Springer Verlag, 1986
- [30] Villemeur A., *Reliability, Availability, Maintainability and Safety Assessment*. Volume 1. Methods and Techniques, Chichester, Wiley, 1992.

#### **Disjointing procedures (or sum of disjoint products methods)**

- [31] Rauzy, A., *Binary Decision Diagrams for Reliability Studies. Pages 381–396, Handbook of Performability Engineering*. K.B. Misra ed., Elsevier. 2008
- [32] Akers B., *Binary decision diagrams*. IEEE Transactions on Computers, Vol. 27, issue 6, pp.509-516, 1978
- [33] Bryant R., *Graph based algorithms for Boolean functions manipulation*. IEEE Transactions on Computers, Vol 35, issue 8, pp.677-691, 1986
- [34] Abraham J.A., *An improved method for network reliability*, IEEE Transactions on Reliability, Vol. 8, No.1, pp.58-61, 1979
- [35] Beichelt F., *Zuverlässigkeit strukturierter Systeme*, Berlin, VEB Verlag Technik, 1988

- [36] Beichelt F., Spross L., *An improved Abraham-method for generating disjoint sums*. *IEEE Transactions on Reliability*, Vol.36, No.1, pp.70-74, 1987
- [37] Heidtmann K.D., *Smaller sums of disjoint products by subproducts inversion*. *IEEE Transactions on Reliability*, Vol.38, No.3, pp.305-311, 1989
- [38] Locks M.O., *Recursive disjoint products. A review of three algorithms*. *IEEE Transactions on Reliability*, Vol. 31, No.1, pp.33-35, 1982
- [39] Locks M.O., *Recent developments in computing of system-reliability*. *IEEE Transactions on Reliability*, Vol. 34, No.5, pp.425-436, 1985
- [40] Locks M.O., *A minimizing algorithm for sum of disjoint products*. *IEEE Transactions on Reliability*, Vol. 36, No.4, pp.445-453, 1987
- [41] Châtelet E., Dutuit Y., Rauzy A and Bouhoufani T. *An optimized procedure to generate sums of disjoint products*, *Reliability Engineering and System Safety*, Vol. 65, No3, pp 280-294, 1999
- [42] Rauzy A., Châtelet E., Dutuit Y. and Bérenguer C. *A practical comparison of methods to assess sum-of-products*, *Reliability Engineering and System Safety*, Vol. 79, No.1, pp 33-42, 2003
- [43] Luo Tong, Trivedi K.S., *An improved algorithm for coherent-system reliability*. *IEEE Transactions on Reliability*, Vol.47, No.1, pp.73-78, 1998
-



# British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

## About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards-based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

## Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at [bsigroup.com/standards](http://bsigroup.com/standards) or contacting our Customer Services team or Knowledge Centre.

## Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at [bsigroup.com/shop](http://bsigroup.com/shop), where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

## Copyright in BSI publications

All the content in BSI publications, including British Standards, is the property of and copyrighted by BSI or some person or entity that owns copyright in the information used (such as the international standardization bodies) and has formally licensed such information to BSI for commercial publication and use.

Save for the provisions below, you may not transfer, share or disseminate any portion of the standard to any other person. You may not adapt, distribute, commercially exploit, or publicly display the standard or any portion thereof in any manner whatsoever without BSI's prior written consent.

## Storing and using standards

Standards purchased in soft copy format:

- A British Standard purchased in soft copy format is licensed to a sole named user for personal or internal company use only.
- The standard may be stored on more than 1 device provided that it is accessible by the sole named user only and that only 1 copy is accessed at any one time.
- A single paper copy may be printed for personal or internal company use only.

Standards purchased in hard copy format:

- A British Standard purchased in hard copy format is for personal or internal company use only.
- It may not be further reproduced – in any format – to create an additional copy. This includes scanning of the document.

If you need more than 1 copy of the document, or if you wish to share the document on an internal network, you can save money by choosing a subscription product (see 'Subscriptions').

## Reproducing extracts

For permission to reproduce content from BSI publications contact the BSI Copyright & Licensing team.

## Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to [bsigroup.com/subscriptions](http://bsigroup.com/subscriptions).

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

**PLUS** is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit [bsigroup.com/shop](http://bsigroup.com/shop).

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email [subscriptions@bsigroup.com](mailto:subscriptions@bsigroup.com).

## Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

## Useful Contacts

### Customer Services

**Tel:** +44 345 086 9001

**Email (orders):** [orders@bsigroup.com](mailto:orders@bsigroup.com)

**Email (enquiries):** [cservices@bsigroup.com](mailto:cservices@bsigroup.com)

### Subscriptions

**Tel:** +44 345 086 9001

**Email:** [subscriptions@bsigroup.com](mailto:subscriptions@bsigroup.com)

### Knowledge Centre

**Tel:** +44 20 8996 7004

**Email:** [knowledgecentre@bsigroup.com](mailto:knowledgecentre@bsigroup.com)

### Copyright & Licensing

**Tel:** +44 20 8996 7070

**Email:** [copyright@bsigroup.com](mailto:copyright@bsigroup.com)

### BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK