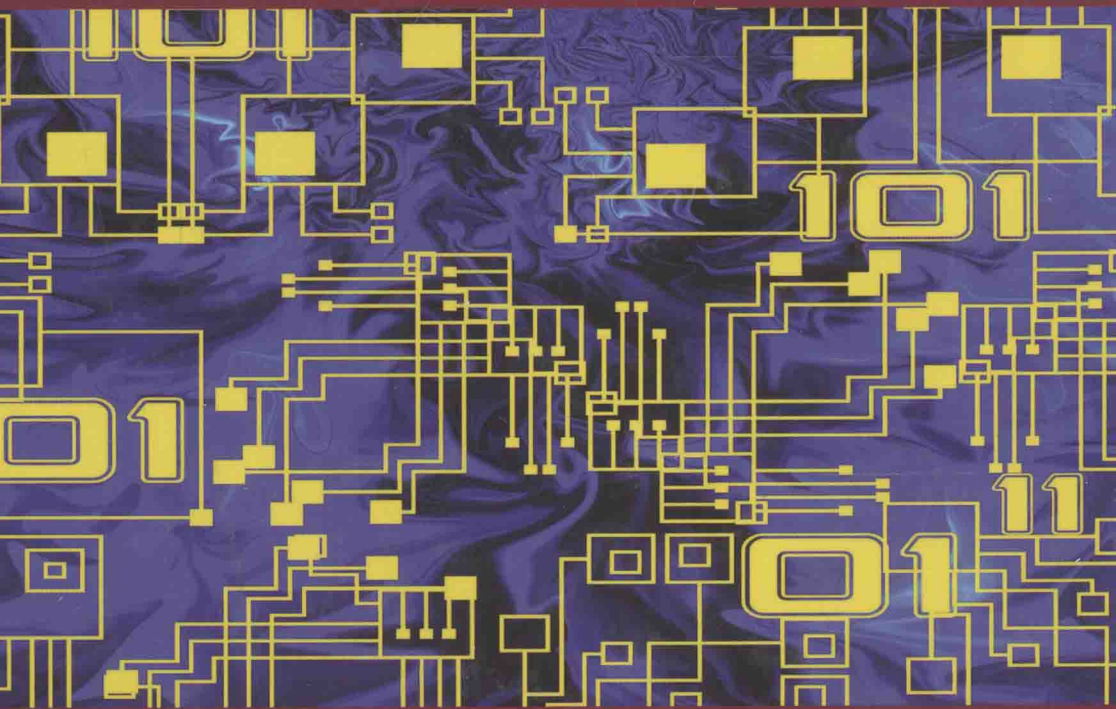


Safety of Computer Architectures

**Edited by
Jean-Louis Boulanger**



ISTE

 **WILEY**

Safety of Computer Architectures

Edited by
Jean-Louis Gaillet



ISTE

 WILEY

First published 2010 in Great Britain and the United States by ISTE Ltd and John Wiley & Sons, Inc.
Adapted and updated from *Sécurisation des architectures informatiques* published 2009 in France by
Hermes Science/Lavoisier © LAVOISIER 2009

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms and licenses issued by the CLA. Enquiries concerning reproduction outside these terms should be sent to the publishers at the undermentioned address:

ISTE Ltd
27-37 St George's Road
London SW19 4EU
UK

www.iste.co.uk

John Wiley & Sons, Inc.
111 River Street
Hoboken, NJ 07030
USA

www.wiley.com

© ISTE Ltd 2010

The rights of Jean-Louis Boulanger to be identified as the author of this work have been asserted by him in accordance with the Copyright, Designs and Patents Act 1988.

Library of Congress Cataloging-in-Publication Data

Sécurisation des architectures informatiques. English

Safety of computer architectures / edited by Jean-Louis Boulanger.

p. cm.

Includes bibliographical references and index.

ISBN 978-1-84821-197-1

1. Computer architecture. 2. Computer systems--Reliability. 3. Computer security. 4. Avionics--Safety measures. I. Boulanger, Jean-Louis. II. Title.

QA76.9.A73S4313 2010

005.8--dc22

2010016489

British Library Cataloguing-in-Publication Data

A CIP record for this book is available from the British Library

ISBN 978-1-84821-197-1

Printed and bound in Great Britain by CPI Antony Rowe, Chippenham and Eastbourne.



Safety of Computer Architectures

Introduction

In recent years, we have experienced an increase in the use of computers and an increase of the inclusion of computers in systems of varying complexity. This evolution affects products of daily life (household appliances, automobiles, etc.) as well as industrial products (industrial control, medical devices, financial transactions, etc.).

The malfunction of systems within these products can have a direct or indirect impact on integrity (injury, pollution, alteration of the environment) and/or on the lives of people (users, population, etc.) or an impact on the functioning of an organization. Processes in industry are becoming increasingly automated. These systems are subject to *dependability* requirements.

Today, dependability has become a *requirement*, not a concern (which was previously the case in high-risk domains such as the nuclear or aerospace industries), in a similar fashion to productivity, which has gradually imposed itself on most industrial and technological sectors.

Dependable systems must protect against certain failures that may have disastrous consequences for people (injury, death), for a company (branding, financial aspects), and/or for the environment.

In the context of systems incorporating “programmed” elements, two types of elements are implemented: hardware elements (computing unit, central processing unit (CPU), memory, bus, field programmable gate array (FPGA), digital signal processor (DSP), programmable logic controller, etc.) and software elements (program, library, operating system, etc.). In this book, we will focus on the safety of the hardware element.

Where the gravity and/or frequency associated with the risks is very important, it is said that the system is “critical”. These “critical” systems are subjected to evaluations (assessment of conformity to standards) and/or certifications (evaluation leading to a certificate of conformity to a standard). This work is carried out by teams that are outside of the realization process.

This book aims to present the principles of securing computer architectures through the presentation of tangible examples.

In Chapter 1 the overall set of techniques (diversity, redundancy, recovery, encoding, etc.) for securing the hardware element of an architecture is presented.

For the railway transport field, Chapters 2, 3, 4, 5 and 11 present the applicable standards (CENELEC EN 50126, EN 50128, and EN 50129) as well as tangible examples (SACEM, SAET-METEOR, CSD, PIPC and the DIGISAFE XME architecture).

Chapters 6 and 7 will cover the field of aeronautics and outer space through three known examples, which are the aircraft from the AIRBUS Company, satellites and the ARIANE 5 launcher. The aviation field was one of the first to establish a referential standard that is currently composed of the DO 178 standard for embedded software development aspects, a trade referential consisting of a set of regulations FAR/JAR, applicable to all aircraft manufacturers and a set of methodological guides produced by the aviation community, ARP 45.74 and ARP 47.61. This referential has been recently complemented by the DO 254 standard, which applies to digital component aspects, such as FPGAs and other ASICs. The DO 278 standard applies to ground software aspects.

For automation-based systems, Chapter 8 presents examples of installations in the oil industry. The IEC 61508 standard allows for a definition and control of the safety objectives (SIL). Chapter 8 presents an opportunity to revisit this standard and its use. This chapter is supplemented by Chapter 10, which is a summary of the implementation of safety instrumented systems (SIS) in industry.

It should be noted that Chapter 12 provides an example of the implementation of a rather interesting automation-based system: the Large Hadron Collider (LHC).

Finally, in Chapter 9 we present examples in the automotive field. The automotive field is currently evolving. This development will result in the establishment of a variation of the IEC 61508 standard for the automotive industry called ISO 26262. This standard takes the safety level concept (called here the automotive safety integrity level, or ASIL) and identifies recommendations for activities and methodologies for implementation in order to achieve a given safety

objective. The automotive field is driven by different types of objectives (cost, place, weight, volume, delays, safety), which requires the establishment of new solutions (see Chapter 9).

It is hoped that this book will enlighten the reader as to the complexity of the systems that are used everyday and the difficulty in achieving a dependable system. It should be noted that this encompasses the need to produce a dependable system but also the need to guarantee the safety during the operational period, which can range from a few days to over 50 years.

Table of Contents

Introduction	xiii
Chapter 1. Principles	1
Jean-Louis BOULANGER	
1.1. Introduction.	1
1.2. Presentation of the basic concepts: faults, errors and failures	1
1.2.1. Obstruction to functional safety	1
1.2.2. Safety demonstration studies	6
1.2.3. Assessment	6
1.3. Safe and/or available architecture.	7
1.4. Resetting a processing unit.	7
1.5. Overview of safety techniques.	8
1.5.1. Error detection	8
1.5.2. Diversity	15
1.5.3. Redundancy	16
1.5.4. Error recovery and retrieval.	42
1.5.5. Partitioning	44
1.6. Conclusion	45
1.7. Bibliography	45
Chapter 2. Railway Safety Architecture	47
Jean-Louis BOULANGER	
2.1. Introduction.	47
2.2. Coded secure processor.	47
2.2.1. Basic principle	47
2.2.2. Encoding	48
2.2.3. Hardware architecture	51
2.2.4. Assessment	52
2.3. Other applications	53

2.3.1. TVM 430	53
2.3.2. SAET-METEOR	54
2.4. Regulatory and normative context	60
2.4.1. Introduction	60
2.4.2. CENELEC and IEC history	63
2.4.3. Commissioning evaluation, certification, and authorization	64
2.5. Conclusion	66
2.6. Bibliography	66
Chapter 3. From the Coded Uniprocessor to 2oo3	69
Gilles LEGOFF and Christophe GIRARD	
3.1. Introduction.	69
3.2. From the uniprocessor to the dual processor with voter.	71
3.2.1. North LGV requirements and the Channel Tunnel	71
3.2.2. The principles of the dual processor with voter by coded uniprocessor.	73
3.2.3. Architecture characteristics	74
3.2.4. Requirements for the Mediterranean LGV	76
3.3. CSD: available safety computer.	80
3.3.1. Background	80
3.3.2. Functional architecture.	82
3.3.3. Software architecture.	85
3.3.4. Synchronization signals	88
3.3.5. The CSD mail system	90
3.4. DIVA evolutions.	93
3.4.1. ERTMS equipment requirements	93
3.4.2. Functional evolution	96
3.4.3. Technological evolution	97
3.5. New needs and possible solutions	99
3.5.1. Management of the partitions.	99
3.5.2. Multicycle services	100
3.6. Conclusion	101
3.7. Assessment of installations	102
3.8. Bibliography	103
Chapter 4. Designing a Computerized Interlocking Module: a Key Component of Computer-Based Signal Boxes Designed by the SNCF	105
Marc ANTONI	
4.1. Introduction.	105
4.2. Issues	107
4.2.1. Persistent bias	107
4.2.2. Challenges for tomorrow	108
4.2.3. Probability and computer safety	109

4.2.4. Maintainability and modifiability	110
4.2.5. Specific problems of critical systems	113
4.2.6. Towards a targeted architecture for safety automats	115
4.3. Railway safety: fundamental notions	116
4.3.1. Safety and availability	116
4.3.2. Intrinsic safety and closed railway world	119
4.3.3. Processing safety	120
4.3.4. Provability of the safety of computerized equipment	121
4.3.5. The signal box	122
4.4. Development of the computerized interlocking module	124
4.4.1. Development methodology of safety systems	125
4.4.2. Technical architecture of the system	130
4.4.3. MEI safety	136
4.4.4. Modeling the PETRI network type	142
4.5. Conclusion	145
4.6. Bibliography	147

Chapter 5. Command Control of Railway Signaling Safety: Safety at Lower Cost

Daniel DRAGO 149

5.1. Introduction	149
5.2. A safety coffee machine	149
5.3. History of the PIPC	150
5.4. The concept basis	155
5.5. Postulates for safety requirements	157
5.6. Description of the PIPC architecture ⁷	159
5.6.1. MCCS architecture	160
5.6.2. Input and output cards	162
5.6.3. Watchdog card internal to the processing unit	169
5.6.4. Head of bus input/output card	170
5.6.5. Field watchdog	171
5.7. Description of availability principles	173
5.7.1. Redundancy	173
5.7.2. Automatic reset	175
5.8. Software architecture	176
5.8.1. Constitution of the kernel	176
5.8.2. The language and the compilers	177
5.8.3. The operating system (OS)	178
5.8.4. The integrity of execution and of data	179
5.8.5. Segregation of resources of different safety level processes	180
5.8.6. Execution cycle and vote and synchronization mechanism	181
5.9. Protection against causes of common failure	186
5.9.1. Technological dissimilarities of computers	186

5.9.2. Time lag during process execution.	187
5.9.3. Diversification of the compilers and the executables	187
5.9.4. Antivalent acquisitions and outputs	187
5.9.5. Galvanic isolation.	188
5.10. Probabilistic modeling	188
5.10.1. Objective and hypothesis	188
5.10.2. Global model.	189
5.10.3. “Simplistic” quantitative evaluation	191
5.11. Summary of safety concepts	194
5.11.1. Concept 1: 2oo2 architecture	194
5.11.2. Concept 2: protection against common modes	195
5.11.3. Concept 3: self-tests	196
5.11.4. Concept 4: watchdog	196
5.11.5. Concept 5: protection of safety-related data	197
5.12. Conclusion	197
5.13. Bibliography	198
Chapter 6. Dependable Avionics Architectures: Example of a Fly-by-Wire system	199
Pascal TRAVERSE, Christine BEZARD, Jean-Michel CAMUS, Isabelle LACAZE, Hervé LEBERRE, Patrick RINGEARD and Jean SOUYRIS	
6.1. Introduction.	199
6.1.1. Background and statutory obligation	200
6.1.2. History	202
6.1.3. Fly-by-wire principles	204
6.1.4. Failures and dependability	204
6.2. System breakdowns due to physical failures	205
6.2.1. Command and monitoring computers	205
6.2.2. Component redundancy	208
6.2.3. Alternatives	212
6.3. Manufacturing and design errors	215
6.3.1. Error prevention.	215
6.3.2. Error tolerance	221
6.4. Specific risks	223
6.4.1. Segregation	223
6.4.2. Ultimate back-up	224
6.4.3. Alternatives	224
6.5. Human factors in the development of flight controls	225
6.5.1. Human factors in design.	225
6.5.2. Human factors in certification	227
6.5.3. Challenges and trends	228
6.5.4. Alternatives	229

6.6. Conclusion	229
6.7. Bibliography	229
Chapter 7. Space Applications	233
Jean-Paul BLANQUART and Philippe MIRAMONT	
7.1. Introduction.	233
7.2. Space system	233
7.2.1. Ground segment.	234
7.2.2. Space segment.	234
7.3. Context and statutory obligation	237
7.3.1. Structure and purpose of the regulatory framework.	237
7.3.2. Protection of space	238
7.3.3. Protection of people, assets, and the environment.	239
7.3.4. Protection of the space system and the mission	241
7.3.5. Summary of the regulatory context	241
7.4. Specific needs	243
7.4.1. Reliability	243
7.4.2. Availability	246
7.4.3. Maintainability	248
7.4.4. Safety	249
7.4.5. Summary	251
7.5. Launchers: the Ariane 5 example	252
7.5.1. Introduction	252
7.5.2. Constraints	253
7.5.3. Object of the avionics launcher.	255
7.5.4. Choice of onboard architecture	256
7.5.5. General description of avionics architecture	257
7.5.6. Flight program	260
7.5.7. Conclusion	281
7.6. Satellite architecture.	281
7.6.1. Overview.	281
7.6.2. Payload	282
7.6.3. Platform	283
7.6.4. Implementation	288
7.6.5. Exploration probes	292
7.7. Orbital transport: ATV example	292
7.7.1. General information	292
7.7.2. Dependability requirements.	295
7.7.3. ATV avionic architecture	296
7.7.4. Management of ATV dependability	299
7.8. Summary and conclusions	302

7.8.1. Reliability, availability, and continuity of service.	302
7.8.2. Safety	304
7.9. Bibliography	304
Chapter 8. Methods and Calculations Relative to “Safety Instrumented Systems” at TOTAL.	307
Yassine CHAABI and Jean-Pierre SIGNORET	
8.1. Introduction.	307
8.2. Specific problems to be taken into account	308
8.2.1. Link between classic parameters and standards’ parameters	308
8.2.2. Problems linked to sawtooth waves	309
8.2.3. Definition	310
8.2.4. Reliability data	312
8.2.5. Common mode failure and systemic incidences.	313
8.2.6. Other parameters of interest.	315
8.2.7. Analysis of tests and maintenance	315
8.2.8. General approach	321
8.3. Example 1: system in 2/3 modeled by fault trees.	322
8.3.1. Modeling without CCF	322
8.3.2. Introduction of the CCF by factor β	323
8.3.3. Influence of test staggering	324
8.3.4. Elements for the calculation of PFH.	326
8.4. Example 2: 2/3 system modeled by the stochastic Petri net	328
8.5. Other considerations regarding HIPS.	333
8.5.1. SIL objectives	333
8.5.2. HIPS on topside facilities	334
8.5.3. Subsea HIPS.	340
8.6. Conclusion	342
8.7. Bibliography	343
Chapter 9. Securing Automobile Architectures.	345
David LIAIGRE	
9.1. Context	345
9.2. More environmentally-friendly vehicles involving more embedded electronics	347
9.3. Mastering the complexity of electronic systems	348
9.4. Security concepts in the automotive field	350
9.4.1. Ensure minimum security without redundancy	350
9.4.2. Hardware redundancy to increase coverage of dangerous failures .	353
9.4.3. Hardware and functional redundancy	358
9.5. Which security concepts for which security levels of the ISO 26262 standard?	364
9.5.1. The constraints of the ISO 26262 standard.	364

9.5.2. The security concepts adapted to the constraints of the ISO 26262 standard	373
9.6. Conclusion	376
9.7. Bibliography	377
Chapter 10. SIS in Industry	379
Grégory BUCHHEIT and Olaf MALASSE	
10.1. Introduction	379
10.2. Safety loop structure.	384
10.2.1. “Sensor” sub-system	386
10.2.2. “Actuator” sub-system	388
10.2.3. Information processing	388
10.2.4. Field networks	401
10.3. Constraints and requirements of the application	407
10.3.1. Programming.	407
10.3.2. Program structure	409
10.3.3. The distributed safety library	411
10.3.4. Communication between the standard user program and the safety program	412
10.3.5. Generation of the safety program.	413
10.4. Analysis of a safety loop	413
10.4.1. Calculation elements	414
10.4.2. PFH calculation for the “detecting” sub-system	416
10.4.3. PFH calculation by “actuator” sub-system	417
10.4.4. PFH calculation by “logic processing” sub-system	418
10.4.5. PFH calculation for a complete loop	419
10.5. Conclusion	423
10.6. Bibliography	424
Chapter 11. A High-Availability Safety Computer	425
Sylvain BARO	
11.1. Introduction	425
11.2. Safety computer	426
11.2.1. Architecture	427
11.2.2. Properties	432
11.3. Applicative redundancy.	433
11.4. Integrated redundancy.	433
11.4.1. Goals	434
11.4.2. Overview of operation	435
11.4.3. Assumptions	435
11.4.4. Hardware architecture	436
11.4.5. Synchronization and time stamping	437
11.4.6. Safety inputs	437

11.4.7. Application and context	438
11.4.8. Safety output.	440
11.4.9. Passivation	443
11.5. Conclusion	443
11.6. Bibliography	446
Chapter 12. Safety System for the Protection of Personnel in the CERN Large Hadron Collider	447
Pierre NININ, Silvia GRAU, Tomasz LADZINSKI and Francesco VALENTINI	
12.1. Introduction	447
12.1.1. Introduction to CERN	447
12.1.2. Legislative and regulatory context	448
12.1.3. Goal of the system	449
12.2. LACS	450
12.3. LASS.	452
12.3.1. IIS Beams.	452
12.3.2. LASS architecture.	454
12.3.3. Performance of safety functions	456
12.3.4. Wired loop	457
12.4. Functional safety methodology	459
12.4.1. Functional safety plan	459
12.4.2. Preliminary risk analysis (PRA)	459
12.4.3. Specification of safety functions	460
12.4.4. Provisional safety study	461
12.4.5. Definitive safety study	465
12.4.6. Verification and validation plan	465
12.4.7. Operation and maintenance plan	465
12.5. Test strategy.	466
12.5.1. Detailed description of the validation process	466
12.5.2. Architecture of the test and the simulation platform.	466
12.5.3. Organization of tests	467
12.5.4. Test platforms	468
12.5.5. Unit validation on site	469
12.5.6. Functional validation on site	471
12.6. Feedback.	472
12.7. Conclusions.	473
12.8. Bibliography	474
Glossary	477
List of Authors	485
Index	487

Chapter 1

Principles

1.1. Introduction

The objective of this chapter¹ is to present the different methods for securing the functional safety of hardware architecture. We shall speak of hardware architecture as safety can be based on one or more calculating units. We shall voluntarily leave aside the “software” aspects.

1.2. Presentation of the basic concepts: faults, errors and failures

1.2.1. *Obstruction to functional safety*

As indicated in [LAP 92], the functional safety of a complex system can be compromised by three types of incidents: failures, faults, and errors. The system elements are subjected to failures, which can potentially result in accidents.

DEFINITION 1.1: FAILURE – as indicated in the IEC 61508 [IEC 98] standard: a failure is the suspension of a functional unit’s ability to accomplish a specified function. Since the completion of a required function necessarily excludes certain behavior, and certain functions can be specified in terms of behavior to avoid, then the occurrence of a behavior to avoid is a failure.

Chapter written by Jean-Louis BOULANGER.

1. This chapter is based on educational material produced together with M. Walter SCHÖN, Professor at the University of Technology of Compiègne, whom I can never thank enough.

From the previous definition, the need to define the concepts of normal (safe) and abnormal (unsafe) conduct can be removed, with a clear boundary between the two.

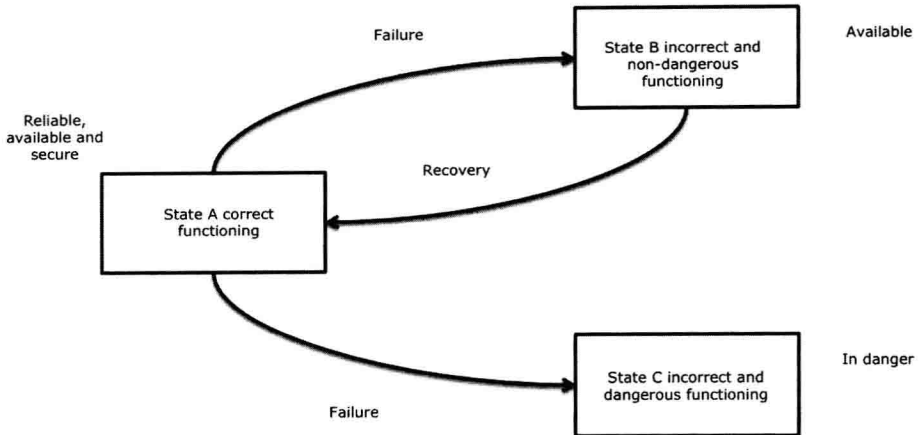


Figure 1.1. *Evolution of the state of the system*

Figure 1.1 shows a representation of the different states of a system (correct, incorrect) and the possible transitions between these states. The system states can be classified into three families:

- correct states: there is no dangerous situation;
- incorrect safe states: a failure was detected and the system is in a safe state;
- incorrect states: this is a dangerous, uncontrolled situation: there are potential accessible accidents.

When the system reaches a fallback state, there may be a partial or complete shutdown of service. The conditions of fallback may allow a return to the correct state after a recovery action.

Failures can be random or systematic. A random failure occurs unpredictably and is the result of damage affecting the hardware aspects of the system. In general, random failure can be quantified because of its nature (wear, aging, etc.).

A systematic failure is linked deterministically to a cause. The cause of the failure can only be eliminated by a reapplication of the production process (design, manufacture, documentation) or by recovery procedures. Given its nature, a systematic failure is not quantifiable.