

Contents

1	Introduction	1
	References	5
2	Hardware Faults	7
2.1	Introduction	7
2.2	Single Event Effects and Other Deviations	9
2.3	Conclusion	10
	References	10
3	Fault Tolerance: Theory and Concepts	11
3.1	Introduction to Reliability Theory	11
3.2	Connection Between Reliability and Fault Tolerance	13
3.3	Models for Fault Tolerance	18
3.4	Chapter Conclusion	21
	References	22
4	Generalized Algorithm of Fault Tolerance (GAFT)	23
4.1	The Generalized Algorithm of Fault Tolerance	24
4.2	Definition of Fault Tolerance by GAFT	28
4.3	Example of Possible GAFT Implementation	28
4.4	GAFT Properties: Performance, Reliability, Coverage	31
4.5	Reliability Evaluation for Fault Tolerance	34
4.6	Hardware Redundancy and Reliability	36
4.6.1	Hardware Redundancy: Reliability Analysis	37
4.7	Conceptual Summary	42
	References	43
5	GAFT Generalization: A Principle and Model of Active System Safety	45
5.1	GAFT Extension: The Method of Active System Safety	47
5.2	GAFT Derivation: A Principle of Active System Safety	47
5.3	Dependency Matrix	48
5.4	Recovery Matrix	50

5.5	PASS Tracing Algorithm	51
5.5.1	Forward Tracing Algorithm	51
5.5.2	Backward Tracing Algorithm.	52
5.6	Chapter Summary.	54
	References.	55
6	System Software Support for Hardware Deficiency: Function and Features.	57
6.1	System Software Life Cycle Versus Fault Tolerance	63
6.2	System Software Phases.	65
	References.	66
7	Testing, Checking, and Hardware Syndrome.	67
7.1	Hardware Checking Process	68
7.2	Analysis of Checking Process	71
7.2.1	The System Model	72
7.2.2	Diagnostic Process Algorithm	73
7.2.3	Procedure T1	74
7.2.4	Extension of the Diagnostic Procedure	77
7.2.5	Testing of Time-Sharing Systems	79
7.2.6	FT Scheduling	82
7.3	System Monitoring of Checking Process: A Syndrome.	83
7.3.1	Access and Location of the Syndrome.	88
7.3.2	Memory Configuration.	90
7.3.3	Interfacing Zone: The Syndrome as Memory Configuration Mechanism	91
7.3.4	Graceful Degradation Approach and Implementation.	92
7.3.5	Reconfiguration of Other Hardware Devices.	96
7.4	Software Support for Hardware Reconfiguration.	96
7.4.1	Software Support for Degradation	96
7.4.2	Hardware Condition Monitor.	98
7.4.3	Hardware Condition Monitor—System Software Support.	100
7.5	Hardware Reconfiguration Outlook.	102
7.6	Summary	103
	References.	104
8	Recovery Preparation	105
8.1	Runtime System Support for Fault Tolerance and Reconfigurability.	105
8.2	Overview of Existing Backward Recovery Techniques.	108
8.2.1	Uncoordinated Recovery Points.	111
8.2.2	Coordinated Recovery Points.	111
8.2.3	Message Logging-Based Rollback Recovery	113
8.2.4	Other Approaches.	115
8.2.5	Implementation Aspects.	115

8.3	Recovery at the Level of Procedures	116
8.4	Hardware Support for Recovery Point Creation	117
8.5	Variable Recovery Point Creation	119
8.5.1	Efficiency Analysis.	120
8.6	Implementation Aspects.	122
8.7	Ultra Reliable Storage	125
8.7.1	Physical Storage Media	126
8.7.2	Flash Properties and Limitations	127
8.7.3	Analysis and Design Considerations	128
8.7.4	Implementation Aspects.	128
8.8	Summary	131
	References.	131
9	Recovery: Searching and Monitoring of Correct Software States.	135
9.1	Recovery as a Process	135
9.2	Modified Linear Algorithm	137
9.2.1	Characteristics of Modified Linear Recovery Algorithm.	138
9.2.2	MLR Execution in Case of Permanent Faults	140
9.2.3	The MLR Algorithm in Case of Several Successive Faults in Case of Permanent Faults	143
9.2.4	Hardware Support for the MLR Algorithm	144
9.3	Summary	145
	References.	145
10	Recovery Algorithms: An Analysis	147
10.1	Comparison of MLR and Two Other Recovery Algorithms of the Same Family	147
10.2	Computational Model.	147
10.3	Linear Recovery Algorithm	149
10.4	Dichotomous Recovery Algorithm.	152
10.5	Modified Linear Recovery	154
10.6	Conclusion on Comparison of Recovery Algorithms	155
10.7	Summary	157
	References.	157
11	Programming Language for Safety Critical Systems	159
11.1	Oberon-07	159
11.2	New Programming Languages Features	160
11.3	Languages Modernization	162
11.3.1	Keywords	162
11.3.2	Enumeration.	162
11.4	Object Orientation	164
11.5	Memory Management	168
11.6	Activities	171
11.6.1	Message Passing Syntax	172
11.6.2	The SYSTEM Resource.	178

11.7	Support for the Three Main GAFT Processes	178
11.7.1	Testing and Checking.	178
11.7.2	Recovery Preparation.	179
11.7.3	Recovery.	180
11.8	Summary	181
	References.	182
12	Proposed Run-Time System Structure	183
12.1	Run-Time System Structure.	183
13	Proposed Run-Time System Versus Existing Approaches.	185
13.1	What Is Available?	185
	References.	187
14	Hardware: The ERRIC Architecture	189
14.1	Processor Architecture	189
14.2	Instruction Set	194
	References.	197
15	Architecture Comparison and Evaluation	199
15.1	Processor Architecture Overview.	199
	References.	205
16	ERRIC Reliability	207
16.1	ERRIC Reliability Analysis	207
16.2	Digging a Bit Deeper in Estimation of Efficiency of Reliability Solutions	210
	References.	211
	Index.	213

Software Design for Resilient Computer Systems

Schagaev, I.; Kaegi-Trachsel, Th.

2016, XIV, 214 p. 70 illus., 51 illus. in color., Hardcover

ISBN: 978-3-319-29463-6