

# Contents

Agraiments	v
Sumari	vii
Sumario	ix
Abstract	xi
1 Introduction	1
1.1 Motivation . . . . .	1
1.2 Objectives . . . . .	4
1.3 Structure of the thesis . . . . .	5
2 Faults modeling	9
2.1 Pathology . . . . .	9
2.2 Manifestation . . . . .	10
2.3 Propagation . . . . .	13
2.4 Modeling . . . . .	15
2.5 Summary . . . . .	18
3 Fault Injection	19
3.1 Introduction . . . . .	19
3.1.1 Fault space: what, where, when . . . . .	20
3.1.2 Properties of fault injection . . . . .	21

3.2 Injection methodologies . . . . .	22
3.2.1 Physical fault injection methods . . . . .	22
3.2.2 Software-based fault injection methods . . . . .	23
3.2.3 Emulation-based fault injection methods . . . . .	23
3.2.4 Simulation-based fault injection methods . . . . .	24
3.2.5 Analysis of injection results . . . . .	24
3.2.6 Summary of methods . . . . .	25
3.3 Injection tools . . . . .	25
3.3.1 Physical fault injection tools . . . . .	26
3.3.2 SWIFI tools . . . . .	26
3.3.3 Emulation-based injection tools . . . . .	27
3.3.4 Simulation-based injection tools . . . . .	28
3.4 The FALLES Tool . . . . .	29
3.4.1 Presentation . . . . .	29
3.4.2 Detailed operation . . . . .	30
3.4.3 Analysis in FALLES . . . . .	34
3.5 Summary . . . . .	36
4 Dependability assessment . . . . .	37
4.1 Introduction . . . . .	37
4.2 Analysis of injection results . . . . .	38
4.3 Multi-level correlation . . . . .	40
4.4 Summary . . . . .	42
5 Fault Tolerance Mechanisms . . . . .	43
5.1 Detection . . . . .	43
5.2 Error handling . . . . .	46
5.3 Fault diagnosis . . . . .	47
5.4 Fault recovery . . . . .	48
5.5 Summary . . . . .	48
6 Discussion and Conclusions . . . . .	51
6.1 Discussion . . . . .	51
6.1.1 Fault models . . . . .	52
6.1.2 Fault injections . . . . .	53

6.1.3 Dependability assessment . . . . .	54
6.1.4 Fault Tolerance mechanisms . . . . .	54
6.1.5 Fault tolerance implementation . . . . .	56
6.2 Conclusion . . . . .	56
6.3 Future work. . . . .	59
7 Summary of contributions	61
7.1 Publications . . . . .	61
7.1.1 Conferences . . . . .	61
7.1.2 Journals . . . . .	63
7.1.3 Book chapters . . . . .	63
7.2 Framework of the Dissertation . . . . .	63
7.2.1 Research projects. . . . .	63
7.2.2 International research stays. . . . .	64
7.2.3 Collaborations. . . . .	64
7.3 Awards . . . . .	65
Appendices	67
A Tolerating multiple faults with proximate manifestations in FPGA-based critical designs for harsh environments	69
A.1 Introduction . . . . .	70
A.2 Faults in SRAM FPGAs . . . . .	71
A.3 Fault tolerance for FPGA-based designs . . . . .	73
A.4 A multiple fault tolerance approach . . . . .	74
A.4.1 Global architecture . . . . .	75
A.4.2 Detailed description . . . . .	75
A.4.3 Design of the FSM controller . . . . .	77
A.4.4 Summary . . . . .	80
A.5 Case study . . . . .	81
A.6 Analysis of results . . . . .	82
A.7 Conclusions . . . . .	84

B The Challenge of Detection and Diagnosis of Fugacious Hardware Faults in VLSI Designs	85
B.1 Introduction	86
B.2 The problem of Fast Fault Detection and Diagnosis	87
B.2.1 On-line detection of faults and errors	88
B.2.2 Considered fault models	90
B.2.3 Fault diagnosis	90
B.3 Solutions for detection and diagnosis	91
B.3.1 Architecture of a faults detection and discrimination system	91
B.3.2 Workflow to apply in the proposed technique	94
B.4 Ongoing Work	95
C Increasing the Dependability of VLSI Systems Through Early Detection of Fugacious Faults	97
C.1 Introduction	98
C.2 Fugacious fault models	100
C.3 Novel architecture for detecting and diagnosing fugacious faults	101
C.4 Proposed implementation flow	106
C.5 First prototype and case study	107
C.6 Results and discussion	109
C.7 Conclusions	112
D An Aspect-oriented Approach to Hardware Fault Tolerance for Embedded Systems	113
D.1 Introduction	114
D.2 Related Work	116
D.2.1 Metaprogramming and aspect orientation	116
D.2.2 Hardware fault and intrusion tolerance automation	117
D.3 Metaprogramming the design of dependable and secure HDL-based embedded systems	119
D.3.1 Open compilation to support the customization of hardware systems	120
D.3.2 Architecting hardware fault tolerance mechanisms as metaprograms	122
D.3.3 Integration within the regular hardware design flow	124
D.4 Dealing with white and black box IP cores as case studies	126
D.4.1 White box IP cores: tolerating transient faults via temporal redundancy	127

D.4.2 Black box IP cores: integrating third party cores for symmetric encryption	130
D.5 Analysis of Results and Discussion	134
D.5.1 Experimental setup	136
D.5.2 Analysis of results	136
D.6 Conclusions and Open Challenges.	139
E Robust communications using automatic deployment of a CRC-generation technique in IP-blocks	143
E.1 Introduction	144
E.2 Research context	145
E.2.1 CRCs and fault tolerance.	145
E.2.2 Metaprograms and open compilation.	146
E.3 CRC as a metaprogram.	147
E.3.1 Phase 1: Infrastructure generation	147
E.3.2 Phase 2: Component encapsulation	149
E.3.3 Phase 3: Component integration	150
E.3.4 Bridging mechanism deployment and VHDL coding	150
E.4 Case study	151
E.4.1 CRC-protected UART transmitter	151
E.4.2 Faultload.	151
E.4.3 Experimental procedure.	152
E.5 Results and discussion	153
E.6 Conclusions.	155
F Towards Certification-aware Fault Injection Methodologies Using Virtual Prototypes	157
F.1 Introduction	158
F.2 Related Work	159
F.3 Certification-Aware Fault Injection in Virtual prototypes	160
F.3.1 Characterizing Fault behaviour at RTL level.	160
F.3.2 Fault injection at Virtual prototypes.	161
F.4 FALLES: Fault injection and Analysis for Low Level Evaluation Suite	162
F.5 Experimental Results	162
F.5.1 Experimental Setup	163
F.5.2 Results	164

F.6 Conclusions . . . . .	165
G Analysis and RTL Correlation of Instruction Set Simulators for Automotive Microcontroller Robustness Verification	167
G.1 Introduction . . . . .	168
G.2 Towards Simulation-based Robustness Verification . . . . .	169
G.2.1 Fault injection at the RTL . . . . .	170
G.2.2 Fault injection at the ISS Level . . . . .	171
G.2.3 ISS-based Verification . . . . .	171
G.3 Correlating RTL with ISS fault injection . . . . .	173
G.4 Experimental Validation . . . . .	175
G.4.1 Experimental Setup . . . . .	175
G.4.2 Experimental Results . . . . .	176
G.5 Related Work . . . . .	180
G.6 Conclusions . . . . .	181
H Characterizing Fault Propagation in Safety-Critical Processor Designs	183
H.1 Introduction . . . . .	184
H.2 Background on Simulation-based Robustness Verification . . . . .	185
H.2.1 Fault injection at the RTL . . . . .	187
H.2.2 Fault injection at the ISS Level . . . . .	187
H.3 Characterizing Fault Propagation . . . . .	188
H.4 Experimental Results . . . . .	190
H.4.1 Experimental Setup . . . . .	190
H.4.2 Results . . . . .	191
H.5 Conclusions . . . . .	194
Bibliography	197