CONTENTS

Contents v
List of Figures viii
List of Tables ix
Preface xiii
Acknowledgments xv
1. INTRODUCTION 1
1.1 Book Motivation and Objectives 1
1.2 Book Organization 4
2. DESIGN OF PROCESSOR-BASED SOC 7
2.1 Integrated Circuits Technology 7
2.2 Embedded Core-Based System-on-Chip Design 8
2.3 Embedded Processors in SoC Architectures 11
3. TESTING OF PROCESSOR-BASED SOC 21
3.1 Testing and Design for Testability 21
3.2 Hardware-Based Self-Testing 32
3.3 Software-Based Self-Testing 41
3.4 Software-Based Self-Test and Test Resource Partitioning 46
3.5 Why is Embedded Processor Testing Important? 48
3.6 Why is Embedded Processor Testing Challenging? 49
4. PROCESSOR TESTING TECHNIQUES 55
4.1 Processor Testing Techniques Objectives 55
4.1.1 External Testing versus Self-Testing 56
4.1.2 DfT-based Testing versus Non-Intrusive Testing 57
4.1.3 Functional Testing versus Structural Testing 58
4.1.4 Combinational Faults versus Sequential Faults Testing 59
4.1.5 Pseudorandom versus Deterministic Testing 60
4.1.6 Testing versus Diagnosis 62
4.1.7 Manufacturing Testing versus On-line/Field Testing 63
4.1.8 Microprocessor versus DSP Testing 63
4.2 Processor Testing Literature 64
4.2.1 Chronological List of Processor Testing Research 64
4.2.2 Industrial Microprocessors Testing 78
4.3 Classification of the Processor Testing Methodologies 78
5. SOFTWARE-BASED PROCESSOR SELF-TESTING 81
5.1 Software-based self-testing concept and flow 82
5.2 Software-based self-testing requirements 87
5.2.1 Fault coverage and test quality 88
5.2.2 Test engineering effort for self-test generation 90

5.2.3 Test application time 91
5.2.4 A new self-testing efficiency measure 96
5.2.5 Embedded memory size for self-test execution 97
5.2.6 Knowledge of processor architecture 98
5.2.7 Component based self-test code development 99
5.3 Software-based self-test methodology overview 100
5.4 Processor components classification 107
5.4.1 Functional components 108
5.4.2 Control components 111
5.4.3 Hidden components 112
5.5 Processor components test prioritization 113
5.5.1 Component size and contribution to fault coverage 115
5.5.2 Component accessibility and ease of test 117
5.5.3 Components' testability correlation 119
5.6 Component operations identification and selection 121
5.7 Operand selection 124
5.7.1 Self-test routine development: ATPG 125
5.7.2 Self-test routine development: pseudorandom 133
5.7.3 Self-test routine development: pre-computed tests 137
5.7.4 Self-test routine development: style selection 139
5.8 Test development for processor components 141
5.8.1 Test development for functional components 141
5.8.2 Test development for control components 141
5.8.3 Test development for hidden components 143
5.9 Test responses compaction in software-based self-testing 146
5.10 Optimization of self-test routines 148
5.10.1 "Chained" component testing 149
5.10.2 "Parallel" component testing 152
5.11 Software-based self-testing automation 153
6. CASE STUDIES – EXPERIMENTAL RESULTS 157
6.1 Parwan processor core 158
6.1.1 Software-based self-testing of Parwan 159
6.2 Plasma/MIPS processor core 160
6.2.1 Software-based self-testing of Plasma/MIPS 163
6.3 Meister/MIPS reconfigurable processor core 168
6.3.1 Software-based self-testing of Meister/MIPS 170
6.4 Jam processor core 171
6.4.1 Software-based self-testing of Jam 172
6.5 oc8051 microcontroller core 173
6.5.1 Software-based self-testing of oc8051 175
6.6 RISC-MCU microcontroller core 176
6.6.1 Software-based self-testing of RISC-MCU 177
6.7 oc54x DSP Core 178
6.7.1 Software-based self-testing of oc54x 179
6.8 Compaction of test responses 181
6.9 Summary of Benchmarks 181
7. PROCESSOR-BASED TESTING OF SOC 185
7.1 The concept 185
7.1.1 Methodology advantages and objectives 188
7.2 Literature review 190
7.3 Research focus in processor-based SoC testing 193
8. CONCLUSIONS 195
References 197
Index 213
About the Authors 217