

Contents

- 1 An Overview About Debugging and Verification Techniques for Embedded Software** 1
 - Djones Lettnin and Markus Winterholer
 - 1.1 The Importance of Debugging and Verification Processes 1
 - 1.2 Debugging and Verification Platforms 4
 - 1.2.1 OS Simulation. 4
 - 1.2.2 Virtual Platform 5
 - 1.2.3 RTL Simulation 5
 - 1.2.4 Acceleration/Emulation 5
 - 1.2.5 FPGA Prototyping 6
 - 1.2.6 Prototyping Board 6
 - 1.2.7 Choosing the Right Platform for Software Development and Debugging 7
 - 1.3 Debugging Methodologies 7
 - 1.3.1 Interactive Debugging 8
 - 1.3.2 Post-Process Debugging 8
 - 1.3.3 Choosing the Right Debugging Methodology 10
 - 1.4 Verification Methodologies 10
 - 1.4.1 Verification Planning 10
 - 1.4.2 Verification Environment Development 11
 - 1.5 Summary 14
 - References. 15
- 2 Embedded Software Debug in Simulation and Emulation Environments for Interface IP** 19
 - Cyprian Wronka and Jan Kotas
 - 2.1 Firmware Debug Methods Overview 19
 - 2.2 Firmware Debuggability 22
 - 2.3 Test-Driven Firmware Development for Interface IP 24
 - 2.3.1 Starting Development 24

2.3.2	First Functional Tests	27
2.3.3	Debugging a System	31
2.3.4	System Performance	33
2.3.5	Interface IP Performance in a Full Featured OS Case	34
2.3.6	Low Level Firmware Debug in a State-of-the-Art Embedded System	35
2.4	Firmware Bring-up as a Hardware Verification Tool	35
2.4.1	NAND Flash	35
2.4.2	xHCI	36
2.5	Playback Debugging with Cadence® Indago™ Embedded Software Debugger	38
2.5.1	Example	39
2.5.2	Coverage Measurement	42
2.5.3	Drawbacks	44
2.6	Conclusions	44
	References	45
3	The Use of Dynamic Temporal Assertions for Debugging	47
	Ziad A. Al-Sharif, Clinton L. Jeffery and Mahmoud H. Said	
3.1	Introduction	47
3.1.1	DTA Assertions Versus Ordinary Assertions	48
3.1.2	DTA Assertions Versus Conditional Breakpoints	50
3.2	Debugging with DTA Assertions	50
3.3	Design	51
3.3.1	Past-Time DTA Assertions	53
3.3.2	Future-Time DTA Assertions	53
3.3.3	All-Time DTA Assertions	54
3.4	Assertion's Evaluation	54
3.4.1	Temporal Cycles and Limits	56
3.4.2	Evaluation Log	57
3.4.3	DTA Assertions and Atomic Agents	57
3.5	Implementation	59
3.6	Evaluation	60
3.6.1	Performance	61
3.7	Challenges and Future Work	62
3.8	Conclusion	63
	References	64
4	Automated Reproduction and Analysis of Bugs in Embedded Software	67
	Hanno Eichelberger, Thomas Kropf, Jürgen Ruf and Wolfgang Rosenstiel	
4.1	Introduction	67
4.2	Overview	69

4.3	Debugger-Based Bug Reproduction	70
4.3.1	State of the Art	71
4.3.2	Theory and Algorithms	73
4.3.3	Implementation	75
4.3.4	Experiments	78
4.4	Dynamic Verification During Replay	80
4.4.1	State of the Art	80
4.4.2	Theory and Workflow	81
4.4.3	Implementation of Assertions During Replay	82
4.4.4	Experiments	83
4.5	Root-Cause Analyses	84
4.5.1	State of the Art	85
4.5.2	Theory and Concepts	86
4.5.3	Implementation	97
4.5.4	Experiments	100
4.6	Summary	104
	References	104
5	Model-Based Debugging of Embedded Software Systems	107
	Padma Iyengar, Elke Pulvermueller, Clemens Westerkamp, Juergen Wuebbelmann and Michael Uelschen	
5.1	Introduction	107
5.1.1	Problem Statement	108
5.1.2	Contribution	109
5.2	Related Work	110
5.3	Model-Based Debugging Framework	112
5.3.1	Overview	112
5.4	Runtime Monitoring	116
5.4.1	Classification of Runtime Monitoring	116
5.4.2	Time-and Memory-Aware Runtime Monitoring Approaches	118
5.5	Experimental Evaluation	119
5.5.1	Software Monitoring	119
5.5.2	On-Chip (Software) Monitoring	123
5.6	Performance Metrics	125
5.6.1	Software Monitoring	125
5.6.2	On-Chip (Software) Monitoring	128
5.7	Discussion and Evaluation	129
5.7.1	Salient Features in the Proposed Approach	130
5.8	Conclusion	131
	References	131

6	A Mechanism for Monitoring Driver-Device Communication	133
	Rafael Melo Macieira and Edna Barros	
6.1	Introduction	133
6.2	Related Works	135
6.3	Proposed Approach	136
6.4	Definition of the HFSM-D State Machine	141
6.5	The TDevC Language	144
6.5.1	TDevC Device Model	144
6.5.2	TDevC Platform Model	150
6.6	Architecture of the Monitoring Module	152
6.7	Experiments and Results	153
6.8	Conclusions	156
6.8.1	Future Works	156
	References	157
7	Model Checking Embedded C Software Using <i>k</i>-Induction and Invariants	159
	Herbert Rocha, Hussama Ismail, Lucas Cordeiro and Raimundo Barreto	
7.1	Introduction	159
7.2	Motivating Example	161
7.3	Induction-Based Verification of C Programs Using Invariants	162
7.3.1	The Proposed <i>k</i> -Induction Algorithm	162
7.3.2	Running Example	167
7.4	Experimental Evaluation	172
7.4.1	Experimental Setup	172
7.4.2	Experimental Results	173
7.5	Related Work	179
7.6	Conclusions	180
	References	181
8	Scalable and Optimized Hybrid Verification of Embedded Software	183
	Jörg Behrend, Djones Lettnin, Alexander Grünhage, Jürgen Ruf, Thomas Kropf and Wolfgang Rosenstiel	
8.1	Introduction	183
8.2	Related Work	184
8.2.1	Contributions	186
8.3	VERIFYR Verification Methodology	186
8.3.1	SPA Heuristic	189
8.3.2	Preprocessing Phase	191
8.3.3	Orchestrator	194

- 8.3.4 Coverage 195
 - 8.3.5 Technical Details 195
 - 8.4 Results and Discussion 197
 - 8.4.1 Testing Environment 197
 - 8.4.2 Motorola Powerstone Benchmark Suite 197
 - 8.4.3 Verification Results Using VERIFYR 199
 - 8.4.4 EEPROM Emulation Software from NEC Electronics 200
 - 8.5 Conclusion and Future Work 203
 - References 203
- Index** 207

Embedded Software Verification and Debugging

Lettnin, D.; Winterholer, M. (Eds.)

2017, XVI, 208 p. 80 illus., 62 illus. in color., Hardcover

ISBN: 978-1-4614-2265-5