Contents

1 General basics 8
1.1 Real-time 8
1.2 Phase driven process model — the V-Model 8
1.3 Build process — from the model to the executable 10
1.4 Summary 18

2 Microprocessor technology basics 20
2.1 Microprocessor design 20
2.2 Code execution 22
2.3 Memory addressing and addressing modes 25
2.4 Wait states and burst accesses 31
2.5 Cache 31
2.6 Pipeline 36
2.7 Interrupts 38
2.8 Traps/exceptions 39
2.9 Data consistency 39
2.10 Comparison of desktop processors versus embedded processors . 41
2.11 Summary 42

3 Operating systems 44
3.1 No OS — endless-loop plus interrupts 44
3.2 OSEK/VDX 47
3.3 Cooperative and preemptive multitasking 53
3.4 POSIX 60
3.5 Summary 66

4 Timing theory 67
4.1 Timing parameters 67
4.2 Statistical aspects 73
4.3 CPU load 77
4.4 Bus load 84
4.5 Logical Execution Time (LET) 85
4.6 Summary 86

5 Timing analysis techniques 88
5.1 Overview, layered view 88
5.2 Definitions of terms 91
5.3 Static code analysis 93
5.4 Code simulation 102
5.5 Timing measurement 110
5.6 Hardware-based tracing 122
5.7 Instrumentation-based tracing 136
5.8 Scheduling simulation 153
5.9 Static scheduling analysis 161
5.10 Optimization using evolutionary algorithms 172

5.11 Timing analysis techniques in the V-Model 174

6 Practical examples of timing problems 177

6.1 Where do all the interrupts come from? 177
6.2 OSEK ECC — rarely the best choice 178
6.3 Rare crashes 17 minutes after reset 181
6.4 Missing or duplicated sensor data 183
6.5 In a race with the handbrake on 188
6.6 Measurement delivers WCET results greater than those from static code analysis 189
6.7 Network management messages appear to soon 191
6.8 Seamless timing process in a mass-production project 192
6.9 Timing analysis saves OEM €12m 193
6.10 Summary 194

7 Multi-core, many-core, and multi-ECU timing 195

7.1 Multi-core basics 195
7.2 Different types of parallel execution 200
7.3 Data consistency, spinlocks 208
7.4 Cloning of memory addresses 213
7.5 Summary 216

8 Timing optimization 218

8.1 Timing optimization at the scheduling level 218
8.2 Timing optimization of memory usage 223
8.3 Timing optimization at code level 228
8.4 Summary and guidelines for timing optimization 247

9 Methodology during the development process 251

9.1 Requirements related to timing 251
9.2 Collaboration during the development 260
9.3 Timing concept, scheduling layout, and OS configuration 262
9.4 Timing debugging 263
9.5 Timing optimization 263
9.6 Timing verification 263
9.7 Early consideration for future functionality 266
9.8 Timing supervision in the final product 267
9.9 Positive example — CoReMa by Vitesco Technologies 269
9.10 Summary 271

10 AUTOSAR 272

10.1 AUTOSAR Classical Platform (CP) 273
10.2 AUTOSAR Adaptive Platform (AP) 276
10.3 TIMEX (AUTOSAR Timing Extensions) 285
10.4 ARTI (AUTOSAR/ASAM Run-Time Interface) 288
10.5 Technical report “Timing Analysis” 291
10.6 Summary 292

11 Safety and ISO 26262 **293**
11.1 Basics 293
11.2 Safety standards, timing, and timing verification 295
11.3 Tools for timing verification 298
11.4 Legal aspects 298
11.5 Summary 299

12 Outlook **300**