

Contents

Preface	xi
Acknowledgments	xvii
1. INTRODUCTION	1
1.1 Terms and scope	1
1.2 Application areas	5
1.3 Growing importance of embedded systems	8
1.4 Structure of this book	9
2. SPECIFICATIONS	13
2.1 Requirements	13
2.2 Models of computation	16
2.3 StateCharts	18
2.3.1 Modeling of hierarchy	19
2.3.2 Timers	23
2.3.3 Edge labels and StateCharts semantics	24
2.3.4 Evaluation and extensions	26
2.4 General language characteristics	27
2.4.1 Synchronous and asynchronous languages	27
2.4.2 Process concepts	28
2.4.3 Synchronization and communication	28

2.4.4	Specifying timing	29
2.4.5	Using non-standard I/O devices	30
2.5	SDL	30
2.6	Petri nets	36
2.6.1	Introduction	36
2.6.2	Condition/event nets	40
2.6.3	Place/transition nets	40
2.6.4	Predicate/transition nets	42
2.6.5	Evaluation	44
2.7	Message Sequence Charts	44
2.8	UML	45
2.9	Process networks	50
2.9.1	Task graphs	50
2.9.2	Asynchronous message passing	53
2.9.3	Synchronous message passing	55
2.10	Java	58
2.11	VHDL	59
2.11.1	Introduction	59
2.11.2	Entities and architectures	60
2.11.3	Multi-valued logic and IEEE 1164	62
2.11.4	VHDL processes and simulation semantics	69
2.12	SystemC	73
2.13	Verilog and SystemVerilog	75
2.14	SpecC	76
2.15	Additional languages	77
2.16	Levels of hardware modeling	79
2.17	Language comparison	82
2.18	Dependability requirements	83

3. EMBEDDED SYSTEM HARDWARE	87
3.1 Introduction	87
3.2 Input	88
3.2.1 Sensors	88
3.2.2 Sample-and-hold circuits	90
3.2.3 A/D-converters	91
3.3 Communication	93
3.3.1 Requirements	94
3.3.2 Electrical robustness	95
3.3.3 Guaranteeing real-time behavior	96
3.3.4 Examples	97
3.4 Processing Units	98
3.4.1 Overview	98
3.4.2 Application-Specific Circuits (ASICs)	100
3.4.3 Processors	100
3.4.4 Reconfigurable Logic	115
3.5 Memories	118
3.6 Output	120
3.6.1 D/A-converters	121
3.6.2 Actuators	122
4. EMBEDDED OPERATING SYSTEMS, MIDDLEWARE, AND SCHEDULING	125
4.1 Prediction of execution times	126
4.2 Scheduling in real-time systems	127
4.2.1 Classification of scheduling algorithms	128
4.2.2 Aperiodic scheduling	131
4.2.3 Periodic scheduling	135
4.2.4 Resource access protocols	140
4.3 Embedded operating systems	143

4.3.1	General requirements	143
4.3.2	Real-time operating systems	144
4.4	Middleware	148
4.4.1	Real-time data bases	148
4.4.2	Access to remote objects	149
5.	IMPLEMENTING EMBEDDED SYSTEMS: HARDWARE/SOFTWARE CODESIGN	151
5.1	Task level concurrency management	153
5.2	High-level optimizations	157
5.2.1	Floating-point to fixed-point conversion	157
5.2.2	Simple loop transformations	159
5.2.3	Loop tiling/blocking	160
5.2.4	Loop splitting	163
5.2.5	Array folding	165
5.3	Hardware/software partitioning	167
5.3.1	Introduction	167
5.3.2	COOL	168
5.4	Compilers for embedded systems	177
5.4.1	Introduction	177
5.4.2	Energy-aware compilation	178
5.4.3	Compilation for digital signal processors	181
5.4.4	Compilation for multimedia processors	184
5.4.5	Compilation for VLIW processors	184
5.4.6	Compilation for network processors	185
5.4.7	Compiler generation, retargetable compilers and design space exploration	185
5.5	Voltage Scaling and Power Management	186
5.5.1	Dynamic Voltage Scaling	186
5.5.2	Dynamic power management (DPM)	189

<i>Contents</i>	ix
5.6 Actual design flows and tools	190
5.6.1 SpecC methodology	190
5.6.2 IMEC tool flow	191
5.6.3 The COSYMA design flow	194
5.6.4 Ptolemy II	195
5.6.5 The OCTOPUS design flow	196
6. VALIDATION	199
6.1 Introduction	199
6.2 Simulation	200
6.3 Rapid Prototyping and Emulation	201
6.4 Test	201
6.4.1 Scope	201
6.4.2 Design for testability	202
6.4.3 Self-test programs	205
6.5 Fault simulation	206
6.6 Fault injection	207
6.7 Risk- and dependability analysis	207
6.8 Formal Verification	209
Bibliography	212
About the author	227
List of Figures	229
Index	237