

# Contents

<b>1</b>	<b>Introduction</b>	1
1.1	History of Terms	1
1.2	Opportunities	4
1.3	Challenges	7
1.4	Common Characteristics	15
1.5	Curriculum Integration of Embedded Systems	16
1.6	Design Flows	19
1.7	Structure of This Book	22
1.8	Problems	24
<b>2</b>	<b>Specifications and Modeling</b>	27
2.1	Requirements	27
2.2	Models of Computation	33
2.3	Early Design Phases	39
2.3.1	Use Cases	39
2.3.2	(Message) Sequence Charts and Time/Distance Diagrams	40
2.3.3	Differential Equations	43
2.4	Communicating Finite State Machines (CFSMs)	44
2.4.1	Timed Automata	45
2.4.2	StateCharts: Implicit Shared Memory Communication	47
2.4.3	Synchronous Languages	56
2.4.4	Message Passing: SDL as an Example	58
2.5	Data Flow	64
2.5.1	Scope	64
2.5.2	Kahn Process Networks	65
2.5.3	Synchronous Data Flow	67
2.5.4	Simulink	71

2.6	Petri Nets . . . . .	72
2.6.1	Introduction . . . . .	72
2.6.2	Condition/Event Nets . . . . .	74
2.6.3	Place/Transition Nets . . . . .	75
2.6.4	Predicate/Transition Nets . . . . .	80
2.6.5	Evaluation. . . . .	81
2.7	Discrete Event-Based Languages . . . . .	83
2.7.1	Basic Discrete Event Simulation Cycle. . . . .	83
2.7.2	Multi-valued Logic . . . . .	84
2.7.3	Transaction-Level Modeling (TLM) . . . . .	89
2.7.4	SpecC . . . . .	91
2.7.5	SystemC™ . . . . .	93
2.7.6	VHDL. . . . .	94
2.7.7	Verilog and SystemVerilog . . . . .	104
2.8	Von-Neumann Languages . . . . .	106
2.8.1	CSP . . . . .	106
2.8.2	Ada. . . . .	107
2.8.3	Java . . . . .	109
2.8.4	Communication Libraries . . . . .	110
2.8.5	Additional Languages. . . . .	112
2.9	Levels of Hardware Modeling . . . . .	112
2.10	Comparison of Models of Computation . . . . .	114
2.10.1	Criteria . . . . .	114
2.10.2	UML™. . . . .	117
2.10.3	Ptolemy II. . . . .	119
2.11	Problems . . . . .	120
<b>3</b>	<b>Embedded System Hardware</b> . . . . .	<b>125</b>
3.1	Introduction . . . . .	125
3.2	Input . . . . .	126
3.2.1	Sensors . . . . .	126
3.2.2	Discretization of Time: Sample-and-Hold Circuits . . . . .	129
3.2.3	Fourier Approximation of Signals. . . . .	129
3.2.4	Discretization of Values: Analog-to-Digital Converters . . . . .	133
3.3	Processing Units . . . . .	139
3.3.1	Application-Specific Circuits (ASICs). . . . .	139
3.3.2	Processors. . . . .	140
3.3.3	Reconfigurable Logic . . . . .	161
3.4	Memories. . . . .	164
3.4.1	Conflicting Goals . . . . .	164
3.4.2	Memory Hierarchies . . . . .	165
3.4.3	Register Files . . . . .	166
3.4.4	Caches . . . . .	167
3.4.5	Scratchpad Memories . . . . .	168

- 3.5 Communication . . . . . 169
  - 3.5.1 Requirements . . . . . 170
  - 3.5.2 Electrical Robustness . . . . . 171
  - 3.5.3 Guaranteeing Real-Time Behavior . . . . . 172
  - 3.5.4 Examples . . . . . 173
- 3.6 Output . . . . . 175
  - 3.6.1 Digital-to-Analog Converters . . . . . 176
  - 3.6.2 Sampling Theorem . . . . . 178
  - 3.6.3 Pulse-width Modulation . . . . . 182
  - 3.6.4 Actuators . . . . . 184
- 3.7 Electrical Energy: Energy Efficiency, Generation, and Storage . . . . . 185
  - 3.7.1 Energy Efficiency of Hardware Components. . . . . 185
  - 3.7.2 Sources of Electrical Energy. . . . . 188
  - 3.7.3 Energy Storage . . . . . 189
- 3.8 Secure Hardware . . . . . 191
- 3.9 Problems . . . . . 193
- 4 System Software . . . . . 197**
  - 4.1 Embedded Operating Systems . . . . . 198
    - 4.1.1 General Requirements. . . . . 198
    - 4.1.2 Real-Time Operating Systems. . . . . 201
    - 4.1.3 Virtual Machines . . . . . 204
  - 4.2 Resource Access Protocols. . . . . 205
    - 4.2.1 Priority Inversion . . . . . 205
    - 4.2.2 Priority Inheritance . . . . . 207
    - 4.2.3 Priority Ceiling Protocol. . . . . 209
    - 4.2.4 Stack Resource Policy . . . . . 211
  - 4.3 ERIKA. . . . . 213
  - 4.4 Embedded Linux . . . . . 216
    - 4.4.1 Embedded Linux Structure and Size. . . . . 217
    - 4.4.2 Real-Time Properties . . . . . 219
    - 4.4.3 Flash Memory File Systems . . . . . 220
    - 4.4.4 Reducing RAM Usage . . . . . 221
    - 4.4.5 uClinux—Linux for MMU-Less Systems . . . . . 222
    - 4.4.6 Evaluating the Use of Linux in Embedded Systems . . . . . 224
  - 4.5 Hardware Abstraction Layers. . . . . 225
  - 4.6 Middleware . . . . . 225
    - 4.6.1 OSEK/VDX COM . . . . . 225
    - 4.6.2 CORBA<sup>®</sup> . . . . . 226
    - 4.6.3 POSIX Threads (Pthreads) . . . . . 227
    - 4.6.4 UPnP, DPWS, and JXTA . . . . . 227
  - 4.7 Real-Time Databases . . . . . 228
  - 4.8 Problems . . . . . 229

- 5 Evaluation and Validation . . . . . 231**
  - 5.1 Introduction . . . . . 231
    - 5.1.1 Scope . . . . . 231
    - 5.1.2 Multi-objective Optimization . . . . . 232
    - 5.1.3 Relevant Objectives . . . . . 234
  - 5.2 Performance Evaluation . . . . . 234
    - 5.2.1 Early Phases . . . . . 235
    - 5.2.2 WCET Estimation. . . . . 235
    - 5.2.3 Real-Time Calculus . . . . . 241
  - 5.3 Quality Metrics . . . . . 244
    - 5.3.1 Approximate Computing. . . . . 244
    - 5.3.2 Simple Criteria of Quality. . . . . 245
    - 5.3.3 Criteria for Data Analysis. . . . . 247
  - 5.4 Energy and Power Models . . . . . 249
    - 5.4.1 General Properties. . . . . 249
    - 5.4.2 Analysis of Memories. . . . . 251
    - 5.4.3 Analysis of Instructions and Inter-Instruction Effects . . . 251
    - 5.4.4 Analysis of Major Functional Processor Units . . . . . 252
    - 5.4.5 Analysis of Processor and Memory Energy Consumption. . . . . 252
    - 5.4.6 Analysis of an Entire Application. . . . . 254
    - 5.4.7 Analysis of Multiple Applications with Multithreading . . . . . 255
    - 5.4.8 Analysis for Communication in an Android Phone. . . . . 256
    - 5.4.9 Worst Case Energy Consumption . . . . . 258
  - 5.5 Thermal Models. . . . . 259
  - 5.6 Dependability and Risk Analysis . . . . . 265
    - 5.6.1 Aspects of Dependability . . . . . 265
    - 5.6.2 Safety Analysis . . . . . 267
    - 5.6.3 Security Analysis . . . . . 268
    - 5.6.4 Reliability Analysis. . . . . 268
    - 5.6.5 Fault Tree Analysis, Failure Mode and Effect Analysis . . . . . 274
  - 5.7 Simulation . . . . . 276
  - 5.8 Rapid Prototyping and Emulation . . . . . 277
  - 5.9 Formal Verification . . . . . 278
  - 5.10 Problems . . . . . 279
- 6 Application Mapping . . . . . 283**
  - 6.1 Definition of Scheduling Problems. . . . . 283
    - 6.1.1 Elaboration on the Design Problem . . . . . 283
    - 6.1.2 Types of Scheduling Problems . . . . . 285
  - 6.2 Scheduling for Uniprocessors. . . . . 292
    - 6.2.1 Independent Jobs . . . . . 292
    - 6.2.2 Scheduling with Precedence Constraints . . . . . 298

- 6.2.3 Periodic Scheduling Without Precedence Constraints . . . 299
- 6.2.4 Periodic Scheduling with Precedence Constraints . . . . . 307
- 6.2.5 Sporadic Events . . . . . 307
- 6.3 Scheduling for Independent Jobs on Identical Multiprocessors . . . . . 307
  - 6.3.1 Partitioned Scheduling of Implicit-Deadline Tasks . . . . . 308
  - 6.3.2 Global Dynamic-Priority Scheduling for Implicit Deadlines . . . . . 311
  - 6.3.3 Global Fixed-Job-Priority Scheduling for Implicit Deadlines . . . . . 313
  - 6.3.4 Global Fixed-Task-Priority Scheduling for Implicit Deadlines . . . . . 316
- 6.4 Dependent Jobs on Homogeneous Multiprocessors . . . . . 318
  - 6.4.1 As-Soon-As-Possible Scheduling . . . . . 319
  - 6.4.2 As-Late-As-Possible Scheduling . . . . . 320
  - 6.4.3 List Scheduling . . . . . 321
  - 6.4.4 Optimal Scheduling with Integer Linear Programming . . . . . 323
- 6.5 Dependent Jobs on Heterogeneous Multiprocessors . . . . . 325
  - 6.5.1 Problem Description . . . . . 325
  - 6.5.2 Static Scheduling with Local Heuristics . . . . . 325
  - 6.5.3 Static Scheduling with Integer Linear Programming . . . . . 329
  - 6.5.4 Static Scheduling with Evolutionary Algorithms . . . . . 329
  - 6.5.5 Dynamic and Hybrid Scheduling . . . . . 334
- 6.6 Problems . . . . . 335
- 7 Optimization . . . . . 337**
  - 7.1 High-Level Optimizations . . . . . 338
    - 7.1.1 Simple Loop Transformations . . . . . 338
    - 7.1.2 Loop Tiling/Blocking . . . . . 340
    - 7.1.3 Loop Splitting . . . . . 342
    - 7.1.4 Array Folding . . . . . 344
    - 7.1.5 Floating-Point to Fixed-Point Conversion . . . . . 345
  - 7.2 Task Level Concurrency Management . . . . . 346
  - 7.3 Compilers for Embedded Systems . . . . . 349
    - 7.3.1 Introduction . . . . . 349
    - 7.3.2 Energy-Aware Compilation . . . . . 350
    - 7.3.3 Memory-Architecture Aware Compilation . . . . . 351
    - 7.3.4 Reconciling Compilers and Timing Analysis . . . . . 359
  - 7.4 Power and Thermal Management . . . . . 361
    - 7.4.1 Dynamic Voltage and Frequency Scaling (DVFS) . . . . . 361
    - 7.4.2 Dynamic Power Management (DPM) . . . . . 364
    - 7.4.3 Thermal Management for MPSoCs . . . . . 364
  - 7.5 Problems . . . . . 365

- 8 Test** ..... 367
  - 8.1 Scope ..... 367
  - 8.2 Test Procedures ..... 368
    - 8.2.1 Test Pattern Generation for Gate Level Models ..... 368
    - 8.2.2 Self-Test Programs ..... 369
  - 8.3 Evaluation of Test Pattern Sets and System Robustness. .... 370
    - 8.3.1 Fault Coverage ..... 370
    - 8.3.2 Fault Simulation ..... 371
    - 8.3.3 Fault Injection. .... 372
  - 8.4 Design for Testability. .... 372
    - 8.4.1 Motivation ..... 372
    - 8.4.2 Scan Design ..... 373
    - 8.4.3 Signature Analysis ..... 374
    - 8.4.4 Pseudo-Random Test Pattern Generation ..... 376
    - 8.4.5 The Built-In Logic Block Observer (BILBO)..... 377
  - 8.5 Problems ..... 378
- Appendix A: Integer Linear Programming** ..... 379
- Appendix B: Kirchhoff’s Laws and Operational Amplifiers** ..... 381
- Appendix C: Paging and Memory Management Units** ..... 387
- References** ..... 389
- Index** ..... 415



<http://www.springer.com/978-3-319-56043-4>

Embedded System Design  
Embedded Systems Foundations of Cyber-Physical  
Systems, and the Internet of Things

Marwedel, P.

2018, XXIV, 423 p. 284 illus., 121 illus. in color.,

Hardcover

ISBN: 978-3-319-56043-4