Contents

PREFACE xv

ACKNOWLEDGMENTS xxi

CHAPTER 1 BASIC REAL-TIME CONCEPTS 1

1.1 Basic Computer Architecture 2
- 1.1.1 Bus Transfer Mechanism 2
- 1.1.2 Input and Output 3
- 1.1.3 Memory 3
- 1.1.4 CPU Operation 4

1.2 Some Terminology 7
- 1.2.1 Software Concepts 7
- 1.2.2 System Concepts 8
- 1.2.3 Real-Time Definitions 9
- 1.2.4 Events and Determinism 11
- 1.2.5 Synchronous and Asynchronous Events 12
- 1.2.6 Determinism 12
- 1.2.7 Time-Loading 13

1.3 Real-Time Design Issues 14

1.4 Example Real-Time Systems 14

1.5 Brief History 16
- 1.5.1 Software 17
- 1.5.2 Hardware 17

1.6 Exercises 18

CHAPTER 2 COMPUTER HARDWARE 19

2.1 CPU 20
- 2.1.1 Addressing Modes 21
- 2.1.2 0-Address Architecture 25
- 2.1.3 1-Address Architecture 31
- 2.1.4 2-Address Architecture 33
- 2.1.5 3-Address Architecture 35
- 2.1.6 Compare, Jump, and Subroutine Instructions 35
- 2.1.7 Interrupt Enable and Disable 38
- 2.1.8 Floating Point Instructions 39
- 2.1.9 Pipelining 39
- 2.1.10 Coprocessors 40
- 2.1.11 RISC Machines 41

2.2 Memories 43
- 2.2.1 Core 44
- 2.2.2 Semiconductor Memories 44
- 2.2.3 Fusible Link 46
- 2.2.4 UVROM 47
- 2.2.5 EEPROM 47
- 2.2.6 Flash Memory 48

2.3 Input and Output 48
- 2.3.1 Programmed I/O 48
- 2.3.2 Memory-Mapped I/O 49
- 2.3.3 DMA 50
- 2.3.4 Interrupt Driven I/O 52

2.4 Other Devices 52
- 2.4.1 MUX Transceivers 52
- 2.4.2 A/D Circuitry 53
- 2.4.3 D/A Circuitry 53
- 2.4.4 Interrupt Controllers 54
- 2.4.5 Watchdog Timer 56

2.5 Exercises 56

CHAPTER 3 LANGUAGE ISSUES 59

3.1 Language Features 60
- 3.1.1 Parameter Passing 60
- 3.1.2 Recursion 64
- 3.1.3 Dynamic Allocation 65
- 3.1.4 Typing
- 3.1.5 Exception Handling 68
- 3.1.6 Abstract Data Typing 68
- 3.1.7 Modularity 71

3.2 Survey of Commonly Used Programming Languages 73
- 3.2.1 BASIC 74
- 3.2.2 FORTRAN 74
- 3.2.3 C 75
- 3.2.4 C++ 78

- 3.2.5 Pascal 78
- 3.2.6 Modula-2 79
- 3.2.7 Ada 80
- 3.2.8 Ada 95 81
- 3.2.9 Assembly Languages 82

3.3 Code Generation 83
- 3.3.1 Know Your Compiler 83

3.4 Schedualability Analysis 84

3.5 Exercises 84

CHAPTER 4 THE SOFTWARE LIFE CYCLE 87

4.1 Phases of the Software Life Cycle 88
- 4.1.1 Concept Phase 89
- 4.1.2 Requirements Phase 89
- 4.1.3 Design Phase 91
- 4.1.4 Programming Phase 94
- 4.1.5 Test Phase 95
- 4.1.6 Maintenance Phase 96

4.2 Nontemporal Transitions in the Software Life Cycle 96

4.3 The Spiral Model 98

4.4 Standards 99
- 4.4.1 DOD-STD-2167A 99
- 4.4.2 ISO 9000 103
- 4.4.3 IEEE 830 104
- 4.4.4 Other Standards 104

4.5 Exercises 107

CHAPTER 5 REAL-TIME SPECIFICATION AND DESIGN TECHNIQUES 109

5.1 Natural Languages 110

5.2 Mathematical Specification 111

5.3 Flowcharts 112

5.4 Structure Charts 113

5.5 Pseudocode and Programming Design Languages 115

5.6 Finite State Automata 117

5.7 Data Flow Diagrams 120
- 5.7.1 DeMarco's Rules 121
- 5.7.2 Hatley and Pribhai's Extensions 124

5.8 Petri Nets 124

5.9 Warnier–Orr Notation 129
- 5.9.1 Indexed Loop 132

5.10 Statecharts 133
5.10.1 Depth 134
5.10.2 Orthogonality
5.10.3 Broadcast Communication 135
5.11 Sanity in Using Graphical Techniques 138
5.12 Exercises 139

CHAPTER 6 REAL-TIME KERNELS 141

6.1 Polled Loop Systems 144
6.1.1 Polled Loop with Interrupts 144
6.2 Phase/State-Driven Code 146
6.3 Coroutines 148
6.4 Interrupt-Driven Systems 150
6.4.1 Context Switching 150
6.4.2 Round-Robin Systems 152
6.4.3 Preemptive Priority Systems 153
6.4.4 Major and Minor Cycles 155
6.4.5 Hybrid Systems 156
6.5 Foreground/Background Systems 156
6.5.1 Background Processing 157
6.5.2 Initialization 158
6.5.3 Real-Time operation 158
6.6 Full-Featured Real-Time Operating Systems 160
6.6.1 Task-Control Block Model 161
6.7 Build or Buy? 164
6.8 POSIX 164
6.9 Exercises 166

CHAPTER 7 INTERTASK COMMUNICATION AND SYNCHRONIZATION 169

7.1 Buffering Data 170
7.1.1 Time-Relative Buffering 170
7.1.2 Ring Buffers 171
7.2 Mailboxes 173
7.2.1 Mailbox Implementation 173
7.2.2 Other Operations on Mailboxes 174
7.2.3 Queues 174
7.3 Critical Regions 175
7.4 Semaphores 175
7.4.1 Mailboxes and Semaphores 177
7.4.2 Counting Semaphores 178
7.4.3 Problems with Semaphores 179
7.4.4 The Test-and-Set Instruction 180

7.5 Event Flags and Signals 181
7.6 Deadlock 183
7.6.1 Avoidance 185
7.6.2 Detect and Recover 186
7.7 Exercises 187

CHAPTER 8 REAL-TIME MEMORY MANAGEMENT 189

8.1 Process Stack Management 190
8.1.1 Task-Control Block Model 190
8.1.2 Managing the Stack 190
8.1.3 Run-Time Ring Buffer 193
8.1.4 Maximum Stack Size 193
8.1.5 Multiple Stack Arrangements 193
8.1.6 Task-Control Block Model 196
8.2 Dynamic Allocation 197
8.2.1 Swapping 198
8.2.2 Overlays 198
8.2.3 MFT 198
8.2.4 MVT 200
8.2.5 Demand Paging 201
8.2.6 Working Sets 203
8.2.7 Real-Time Garbage Collection 204
8.2.8 Contiguous File Systems 204
8.3 Static Schemes 205
8.4 Exercises 205

CHAPTER 9 SYSTEM PERFORMANCE ANALYSIS AND OPTIMIZATION 207

9.1 Response-Time Calculation 208
9.1.1 Polled Loops 208
9.1.2 Coroutines/Phase-Driven Code 208
9.1.3 Interrupt Systems 209
9.2 Interrupt Latency 210
9.2.1 Propagation Delay 211
9.2.2 Macroinstruction Execution Times 211
9.2.3 Interrupts Disabled 211
9.2.4 Preemption 212
9.2.5 Low Priority Interrupts High 212
9.3 Time-Loading and Its Measurement 212
9.3.1 Using a Logic Analyzer 213
9.3.2 Instruction Counting 213
9.3.3 Pictorial Representation 216
9.3.4 Instruction Execution Time Simulators 217
9.3.5 Deterministic Performance 218
9.4 Scheduling Is NP-Complete 218

9.5 Reducing Response Times and Time-Loading 219
9.5.1 Compute at Slowest Cycle 220
9.5.2 Scaled Arithmetic 220
9.5.3 Binary Angular Measurement 221
9.5.4 Look-Up Tables 222
9.5.5 Basic Optimization Theory 223
9.5.6 Other Optimization Techniques 232
9.5.7 Combination Effects 233
9.5.8 Speculative Execution 234
9.6 Analysis of Memory Requirements 234
9.6.1 Memory-Mapped I/O and DMA Memory 235
9.6.2 Program Area 236
9.6.3 RAM Area 236
9.6.4 Stack Area 236
9.6.5 Memory Management Schemes 237
9.7 Reducing Memory-Loading 237
9.7.1 Variable Selection 237
9.7.2 Reuse Variables 238
9.7.3 Memory Fragamentation 238
9.7.4 Self-Modifying Code 238
9.8 I/O Performance 239
9.9 Exercises 239

CHAPTER 10 QUEUING MODELS 241

10.1 Probability Functions 241
10.1.1 Continuous 242
10.2 Discrete 243
10.3 Basic Buffer Size Calculation 245
10.3.1 Handling Bursts of Data 245
10.3.2 Variable Buffer Size Calculation 246
10.4 Classical Queuing Theory 248
10.4.1 The M/M/1 Queue 249
10.4.2 Service and Production Rates 250
10.4.3 More Buffer Calculations 251
10.4.4 Response-Time Modeling 252
10.4.5 Other Queuing Models 252
10.5 Little's Law 253
10.6 Erlang's Formula 253
10.7 Exercises 254

CHAPTER 11 RELIABILITY, TESTING, AND FAULT TOLERANCE 255

11.1 Faults, Failures, Bugs, and Effects 255
11.2 Reliability 256
11.2.1 Formal Definition 256
11.2.2 Calculating System Reliability 258

11.3 Testing 263
11.3.1 Unit Level Testing 264
11.3.2 System-Level Testing 267
11.3.3 Statistically Based Testing 267
11.3.4 Cleanroom Testing 268
11.3.5 Stress Testing 269
11.4 Fault Tolerance 269
11.4.1 General Problems Handling 269
11.4.2 N-Version Programming 271
11.4.3 Built-In-Test Software 271
11.4.4 CPU Testing 271
11.4.5 Memory Testing 272
11.4.6 Spurious and Missed Interrupts 276
11.4.7 Dealing with Bit Failures 277
11.5 Exercises 278

CHAPTER 12 MULTIPROCESSING SYSTEMS 281

12.1 Classification of Architectures 282
12.2 Distributed Systems 283
12.2.1 Embedded 283
12.2.2 Organic 283
12.2.3 System Specification 284
12.2.4 Reliability in Distributed Systems 286
12.2.5 Calculation of Reliability in Distributed Systems 286
12.2.6 Increased Reliability in Distributed Systems 289
12.3 Non-Von Neuman Architectures 291
12.3.1 Data Flow Architectures 292
12.3.2 Systolic Processors 294
12.3.3 Wavefront Processors 297
12.3.4 Transputers 298
12.4 Exercises 299

CHAPTER 13 HARDWARE/SOFTWARE INTEGRATION 301

13.1 Goals of Real-Time System Integration 302
13.1.1 System Unification 302
13.1.2 System Validation 303
13.2 Tools 303
13.2.1 Multimeters 304
13.2.2 Oscilloscope 304
13.2.3 Logic Analyzer 304
13.2.4 In-Circuit Emulator 305
13.2.5 Software Simulators 306
13.2.6 Hardware Prototypes/Simulators 307
13.2.7 Debuggers 307

13.3 Methodology 307
13.3.1 Establishing a Baseline 307
13.3.2 Backoff Method 308
13.3.3 Patching 309
13.4 The Software Heisenberg Uncertainty Principle 310
13.4.1 Real-World Analogies 310
13.4.2 The Software Heisenberg Uncertainty Principle 311
13.4.3 Testing of Software 311
13.4.4 Time- and Memory-Loading 312
13.4.5 Other Implications 312
13.5 Exercises 312

CHAPTER 14 REAL-TIME APPLICATIONS 315

14.1 Real-Time Systems as Complex Systems 315
14.2 The First Real-Time Application 316
14.3 Real-Time Databases 317
14.4 Real-Time Image Processing 319
14.4.1 Virtual Reality 320
14.4.2 Multimedia 321
14.5 Real-Time UNIX 324
14.6 Building Real-Time Applications with Real-Time Programming Languages 324
14.7 Exercises 326

GLOSSARY 327

BIBLIOGRAPHY 345

INDEX 355

ABOUT THE AUTHOR 361