Contents

Foreword to the First Edition xv
Preface to the First Edition xvii
Preface to the Second Edition xxi
Preface to the Third Edition xxiii

CHAPTER 1 Embedded Computing 1
1.1 Introduction 1
1.2 Complex systems and microprocessors 1
1.2.1 Embedding computers 2
1.2.2 Characteristics of embedded computing applications 4
1.2.3 Why use microprocessors? 6
1.2.4 Cyber-physical systems 7
1.2.5 Challenges in embedded computing system design 8
1.2.6 Performance of embedded computing systems 9
1.3 The embedded system design process 10
1.3.1 Requirements 12
1.3.2 Specification 16
1.3.3 Architecture design 17
1.3.4 Designing hardware and software components 19
1.3.5 System integration 19
1.3.6 Formalisms for system design 20
1.3.7 Structural description 21
1.3.8 Behavioral description 25
1.4 Design example: Model train controller 28
1.4.1 Requirements 29
1.4.2 DCC 30
1.4.3 Conceptual specification 32
1.4.4 Detailed specification 35
1.4.5 Lessons learned 42
1.5 A guided tour of this book 42
1.5.1 Chapter 2: Instruction sets 43
1.5.2 Chapter 3: CPUs 43
1.5.3 Chapter 4: Computing platforms 44
1.5.4 Chapter 5: Program design and analysis 44
1.5.5 Chapter 6: Processes and operating systems 45
1.5.6 Chapter 7: System design techniques 46
1.5.7 Chapter 8: Networks and multiprocessors 47

1.6 Summary 47
What we learned 48
Further reading 48
Questions 48
Lab exercises 50

CHAPTER 2 Instruction Sets 51
2.1 Introduction 51
2.2 Preliminaries 51
2.2.1 Computer architecture taxonomy 52
2.2.2 Assembly languages 54
2.2.3 VLIW processors 56
2.3 ARM processor 57
2.3.1 Processor and memory organization 58
2.3.2 Data operations 59
2.3.3 Flow of control 66
2.3.4 Advanced ARM features 72
2.4 PICmicro mid-range family 73
2.4.1 Processor and memory organization 73
2.4.2 Data operations 73
2.4.3 Flow of control 76
2.5 TI C55x DSP 77
2.5.1 Processor and memory organization 78
2.5.2 Addressing modes 81
2.5.3 Data operations 83
2.5.4 Flow of control 84
2.5.5 C coding guidelines 86
2.6 TI C64x 87
2.7 Summary 90
What we learned 90
Further reading 91
Questions 91
Lab exercises 93

CHAPTER 3 CPUs 95
3.1 Introduction 95
3.2 Programming input and output 96
3.2.1 Input and output devices 96
3.2.2 Input and output primitives 98
3.2.3 Busy-wait I/O 99
3.2.4 Interrupts 101

3.3 Supervisor mode, exceptions, and traps 114
3.3.1 Supervisor mode 114
3.3.2 Exceptions 115
3.3.3 Traps 115
3.4 Co-processors 115
3.5 Memory system mechanisms 116
3.5.1 Caches 116
3.5.2 Memory management units and address translation 123
3.6 CPU performance 128
3.6.1 Pipelining 128
3.6.2 Cache performance 132
3.7 CPU power consumption 133
3.8 Design example: Data compressor 137
3.8.1 Requirements and algorithm 137
3.8.2 Specification 140
3.8.3 Program design 141
3.8.4 Testing 148
3.9 Summary 149
What we learned 149
Further reading 150
Questions 150
Lab exercises 153

CHAPTER 4 Computing Platforms 155
4.1 Introduction 155
4.2 Basic computing platforms 155
4.2.1 Platform hardware components 156
4.2.2 Platform software components 158
4.3 The CPU bus 159
4.3.1 Bus organization and protocol 160
4.3.2 DMA 167
4.3.3 System bus configurations 170
4.4 Memory devices and systems 172
4.4.1 Memory system organization 174
4.5 Designing with computing platforms 176
4.5.1 Example platforms 176
4.5.2 Choosing a platform 176
4.5.3 Intellectual property 179
4.5.4 Development environments 180
4.5.5 Debugging techniques 181
4.5.6 Debugging challenges 183

4.6 Consumer electronics architecture 185
4.6.1 Consumer electronics use cases and requirements ... 185
4.6.2 File systems 187
4.7 Platform-level performance analysis 188
4.8 Design example: Alarm clock 193
4.8.1 Requirements 193
4.8.2 Specification 194
4.8.3 System architecture 197
4.8.4 Component design and testing 200
4.8.5 System integration and testing 200
4.9 Design example: Audio player 200
4.9.1 Theory of operation and requirements 200
4.9.2 Specification 202
4.9.3 System architecture 204
4.9.4 Component design and testing 206
4.9.5 System integration and debugging 206
4.10 Summary 207
What we learned 207
Further reading 207
Questions 207
Lab exercises 210

CHAPTER 5 Program Design and Analysis 213
5.1 Introduction 213
5.2 Components for embedded programs 214
5.2.1 State machines 214
5.2.2 Circular buffers and stream-oriented programming ..216
5.2.3 Queues and producer/consumer systems 221
5.3 Models of programs 223
5.3.1 Data flow graphs 224
5.3.2 Control/data flow graphs 226
5.4 Assembly, linking, and loading 228
5.4.1 Assemblers 229
5.4.2 Linking 233
5.4.3 Object code design 235
5.5 Compilation techniques 236
5.5.1 The compilation process 236
5.5.2 Basic compilation methods 237
5.5.3 Compiler optimizations 245
5.6 Program-level performance analysis 254
5.6.1 Elements of program performance 256
5.6.2 Measurement-driven performance analysis 259

5.7 Software performance optimization 262
5.7.1 Loop optimizations 262
5.7.2 Cache optimizations 264
5.7.3 Performance optimization strategies 265
5.8 Program-level energy and power analysis and optimization 266
5.9 Analysis and optimization of program size 270
5.10 Program validation and testing 271
5.10.1 Clear-box testing 271
5.10.2 Black-box testing 278
5.10.3 Evaluating functional tests 279
5.11 Design example: Software modem 280
5.11.1 Theory of operation and requirements 280
5.11.2 Specification 283
5.11.3 System architecture 283
5.11.4 Component design and testing 284
5.11.5 System integration and testing 285
5.12 Design example: Digital still camera 285
5.12.1 Theory of operation and requirements 285
5.12.2 Specification 290
5.12.3 System architecture 293
5.12.4 Component design and testing 296
5.12.5 System integration and testing 296
5.13 Summary 296
What we learned 296
Further reading 297
Questions 297
Lab exercises 305

CHAPTER 6 Processes and Operating Systems 307
6.1 Introduction 307
6.2 Multiple tasks and multiple processes 308
6.2.1 Tasks and processes 308
6.3 Multirate systems 310
6.3.1 Timing requirements on processes 311
6.3.2 CPU usage metrics 316
6.3.3 Process state and scheduling 316
6.3.4 Running periodic processes 317
6.4 Preemptive real-time operating systems 319
6.4.1 Two basic concepts 320
6.4.2 Processes and context 321
6.4.3 Processes and object-oriented design 324

6.5 Priority-based scheduling 325
6.5.1 Rate-monotonic scheduling 326
6.5.2 Shared resources 330
6.5.3 Priority inversion 332
6.5.4 Earliest-deadline-first scheduling 333
6.5.5 RMS versus EDF 337
6.5.6 A closer look at our modeling assumptions 337
6.6 Interprocess communication mechanisms 340
6.6.1 Shared memory communication 340
6.6.2 Message passing 341
6.6.3 Signals 342
6.6.4 Mailboxes 343
6.7 Evaluating operating system performance 344
6.8 Power optimization strategies for processes 349
6.9 Example real-time operating systems 352
6.9.1 POSIX 352
6.9.2 Windows CE 357
6.10 Design example: Telephone answering machine 361
6.10.1 Theory of operation and requirements 361
6.10.2 Specification 364
6.10.3 System architecture 366
6.10.4 Component design and testing 368
6.10.5 System integration and testing 368
6.11 Design example: Engine control unit 369
6.11.1 Theory of operation and requirements 369
6.11.2 Specification 370
6.11.3 System architecture 371
6.11.4 Component design and testing 373
6.11.5 System integration and testing 374
6.12 Summary 374
What we learned 374
Further reading 374
Questions 375
Lab exercises 380

CHAPTER 7 System Design Techniques 381

7.1 Introduction 381
7.2 Design methodologies 381
7.2.1 Why design methodologies? 381
7.2.2 Design flows 383

7.3 Requirements analysis 389
7.4 Specifications 390
7.4.1 Control-oriented specification languages 391
7.4.2 Advanced specifications 394
7.5 System analysis and architecture design 396
7.5.1 CRC cards 396
7.6 Quality assurance 400
7.6.1 Quality assurance techniques 402
7.6.2 Verifying the specification 404
7.6.3 Design reviews 406
7.7 Summary 407
What we learned 407
Further reading 408
Questions 408
Lab exercises 408

CHAPTER 8 Networks and Multiprocessors 409
8.1 Introduction 409
8.2 Why networks and multiprocessors? 409
8.3 Categories of multiprocessors 412
8.4 Distributed embedded systems 414
8.4.1 Network abstractions 414
8.4.2 CAN bus 416
8.4.3 Distributed computing in cars and airplanes 419
8.4.4 I²C bus 422
8.4.5 Ethernet 426
8.4.6 Internet 429
8.5 MPSoCs and shared memory multiprocessors 431
8.5.1 Heterogeneous shared memory multiprocessors 431
8.5.2 Accelerators 432
8.5.3 Accelerator performance analysis 434
8.5.4 Scheduling and allocation 438
8.6 Design example: Video accelerator 441
8.6.1 Video compression 441
8.6.2 Algorithm and requirements 443
8.6.3 Specification 445
8.6.4 Architecture 446
8.6.5 Component design 449
8.6.6 System testing 449
8.7 Application example: Compact disc 449

8.8 Summary 454
What we learned 454
Further reading 454
Questions 455
Lab exercises 456

Glossary **459**
References **477**
Index **487**