# Contents

**Preface** xiii

**Chapter 1 Introduction** 1

1.1 What is an Embedded System? 1
1.2 What's Unique About the Design Goals for Embedded Software? 3
1.3 What Does "Real-Time" Mean? 5
1.4 What Does "Multitasking" Mean? 6
1.5 How Powerful Are Embedded Processors? 7
1.6 What Programming Languages Are Used? 7
1.7 What Is a "Real-Time Kernel"? 8
1.8 How Is Building an Embedded Application Unique? 9
1.9 How Big Are Typical Embedded Programs? 11
1.10 The Software Used in This Book 12
Problems 14

**Chapter 2 Data Representation** 15

2.1 Fixed-Precision Binary Numbers 15
2.1.1 Positional Number Systems 16
2.1.2 Binary-to-Decimal Conversion 17
2.1.3 Decimal-to-Binary Conversion 17
2.1.4 Counting 19
2.1.5 Fixed Precision and Rollover 19
2.1.6 Hexadecimal Representation 20

2.2 Binary Representation of Integers 21
2.2.1 Signed Integers 21
2.2.2 Positive and Negative Representations of the Same Magnitude 22
2.2.3 Interpreting the Value of a 2's-Complement Number 23
2.2.4 More on Range and Overflow 24
2.2.5 2's Complement and Hardware Complexity 25

2.3 Binary Representation of Real Numbers 28
2.3.1 Fixed-Point Representation 28
2.3.2 Fixed-Point Using a Universal 16.16 Format 30
2.3.3 Fixed-Point Using a Universal 32.32 Format 32
2.3.4 Floating-Point Representation 35
4.4.4 Registers 75
4.4.5 The Stack 77
4.5 The Intel Real Mode Architecture 78
   4.5.1 Segmented Addressing 79
   4.5.2 Addressing Modes 81
4.6 The Intel Protected Mode Architecture 83
   4.6.1 Segment Registers and The Global Descriptor Table 84
   4.6.2 The Flat Memory Model 85
   4.6.3 Addressing Modes 85
4.7 Operand and Address-Size Override Prefixes 86
4.8 The Intel Data Manipulation Instructions 86
   4.8.1 Data Movement, Stack, and I/O Instructions 87
   4.8.2 Arithmetic Instructions 89
   4.8.3 Bitwise Instructions 91
   4.8.4 Shift Instructions 91
Problems 93

Chapter 5 Mixing C and Assembly 96
5.1 Programming in Assembly 96
5.2 Register Usage Conventions 98
5.3 Typical Use of Addressing Options 98
   5.3.1 Accessing Data Whose Address is a Constant 99
   5.3.2 Accessing Data Whose Address is a Variable 100
5.4 Instruction Sequencing 101
   5.4.1 Compound Conditionals 102
   5.4.2 If-Then-Else Statements 104
   5.4.3 Building Loops 105
   5.4.4 Faster Loops with String Instructions 106
5.5 Procedure Call and Return 107
5.6 Parameter Passing 108
5.7 Retrieving Parameters 110
5.8 Everything is Pass by Value 112
5.9 Temporary Variables 112
Problems 115

Chapter 6 Input/Output Programming 117
6.1 The Intel I/O Instructions 118
6.2 Synchronization, Transfer Rate, and Latency 118
6.3 Polled Waiting Loops 119
Appendix D: Programming Projects 225
Files Required from the CD-ROM for All Applications 225
Files Required for Nonpreemptive Multithreaded Applications 225
Files Required for Preemptive Multithreaded Applications 226
Compiling and Assembling Your Embedded Application 226
Linking Your Embedded Application 226
Preparing the Boot Diskette 227
Running Your Embedded Application 227
Program 1: Getting Started with the DJGPP Compiler Tools 228
Program 2: Using Fixed-Point Real Numbers 230
Program 3: Using Macros and Packed Operands 231
Program 4: Using “Makefiles” 232
Program 5: Coding Extended Precision Multiplication in Assembly 235
Program 6: Coding Extended Precision Division in Assembly 237
Program 7: Polled Waiting Loop and Interrupt-Driven I/O 238
Program 8: A Simple Nonpreemptive Multithreaded Application 240
Program 9: Preemptive Kernels and Shared Resources 242
Program 10: Avoiding Unbounded Priority Inversion 245
Program 11: Avoiding Deadlock 246

Appendix E: The libepc Library 247
Memory Layout and Initialization 247
Display Functions (display.c) 248
Window Functions (window.c) 250
Keyboard Functions (keyboard.c) 251
Speaker Functions (speaker.c) 252
Timer Functions (timer.c, cycles.asm) 252
Interrupt Vector Access Functions (init-idt.c) 253
Dynamic Memory Allocation Functions (heap.c) 254
Fixed Point (fixedpt.asm) 254
Interfunction Jumps (setjmp.asm) 255
Miscellaneous Functions (init-crt.c) 256

Appendix F: The Boot Loader 257

Index 258