Contents

Preface *xv*

1. Introduction to Embedded Systems **1-34**

1.1 Embedded System *1*
1.2 Processor Embedded into a System *4*
1.3 Embedded Hardware Units and Devices in a System *11*
1.4 Embedded Software in a System and an Overview of Programming Languages *16*
1.5 Introduction to Embedded-system design *21*
1.6 Introduction to Embedded-system Architecture *22*
1.7 Introduction to Embedded-system Model *23*
1.8 Classification of Embedded Systems *25*
1.9 Skills required for an Embedded-system Designer *26*
1.10 Examples of the Embedded Systems *27*
Summary *28*
Keywords *29*
Review Questions *33*
Practice Exercises *34*

2. Embedded Systems Design and Development Process **35-70**

2.1 Embedded System-On-Chip (SoC) and Use of VLSI Circuit Design Technology *36*
2.2 Complex Systems Design and Processors *40*
2.3 Build Process for Embedded Systems *45*
2.4 Design Process in Embedded System *45*
2.5 Design Challenges in Embedded-System Design *50*
2.6 Challenges in Embedded-System Design: Optimising the Design Metrics *51*
2.7 Challenges and Issues Related to Embedded-Software Development *53*
2.8 Hardware–Software Co-Design in an Embedded System *55*
2.9 Embedded-System Design Technologies *57*
2.10 Formalism of System Design *59*
2.11 Design Process and Design Examples *59*
Summary *67*
Keywords *68*
Review Questions *69*
Practice Exercises *70*

3. 8051, AVR and ARM Microcontrollers, Real-World Interfacing, and the Inputs and Outputs Using Buses 71-109

3.1 Introduction to Microcontrollers and Microprocessors *72*
3.2 Embedded Versus External Memory Devices *72*
3.3 Example of a Microcontroller–8051 Architecture *73*
3.4 ATMEL AVR Microcontrollers *82*
3.5 ARM Microcontrollers *84*
3.6 Computer-System Buses *85*
3.7 Real-World Interfacing *91*
3.8 I/O Performance *97*
3.9 I/O Buses *98*
3.10 Network-Oriented Bus Arbitration *101*
3.11 Buses *102*
3.12 Multilevel Buses *102*
Summary 105
Keywords 105
Review Questions 108
Practice Exercises 109

4. Introduction to Advanced Architectures and Processor-Memory Organisations 110-143

4.1 Processor and Memory Organisation *110*
4.2 Introduction to Advanced Processor Architectures *114*
4.3 Processor Organisation *118*
4.4 Instruction-Level Parallelism *120*
4.5 Intel x86 Architecture (8086, 80386, 80486 and Pentium) *122*
4.6 ARM *124*
4.7 SHARC *126*
4.8 Memory Types and Addresses *127*
4.9 Memory Addresses *134*
4.10 Memory Hierarchy and Cache *135*
4.11 Performance Metrics *136*
4.12 Selection of Processor and Memory Devices *137*
Summary 139
Keywords 139
Review Questions 142
Practice Exercises 142

5. I/O Devices, Communication Buses and Distributed Networked Embedded Architectures 144-192

5.1 I/O Types and Examples *145*
5.2 Serial Communication Devices *150*
5.3 Parallel Device Ports *159*

5.4 Sophisticated Interfacing Features in Device Ports *163*
5.5 Wireless Devices *164*
5.6 Timer and Counting Devices *164*
5.7 Distributed Network Embedded Systems Architecture *167*
5.8 Serial Bus Communication Protocols *169*
5.9 Parallel Bus Device Protocols—Parallel Communication Network Using the ISA, PCI, PCI-X and Advanced Buses *175*
5.10 Internet Enabled Systems—Network Protocols *178*
5.11 Wireless and Mobile System Protocols *182*
Summary *186*
Keywords *187*
Review Questions *191*
Practice Exercises *191*

6. Device Drivers and Interrupts Service Mechanism 193-226

6.1 Port for Device Accessess without Interrupts Servicing Mechanism Using Programmed I/O *193*
6.2 Interrupt-driven Input and Output *199*
6.3 Interrupt Service Routine Concept *201*
6.4 Interrupt Sources *202*
6.5 Hardware Interrupts *203*
6.6 Software Interrupts *204*
6.7 Interrupt-servicing Mechanism *207*
6.8 Multiple Interrupts *211*
6.9 Interrupt Service Threads as Second-Level Interrupt Handlers *213*
6.10 Context and the Periods for Context Switching *214*
6.11 Interrupt Latency *214*
6.12 Interrupt-Service Deadline *216*
6.13 Classification of Processors' Interrupt-Service Mechanism from Context-Saving Angle *217*
6.14 Direct Memory Access Driven I/O *217*
6.15 Device Driver Programming *220*
Summary *223*
Keywords *224*
Review Questions *226*
Practice Exercises *226*

7. Programming Concepts and Embedded Programming in C, C++ and Java 227-262

7.1 Programming in Assembly Language (ALP) and in High-Level Language 'C' *228*
7.2 'C' Program Elements: Header and Source Files and Preprocessor Directives *229*

7.3 Program Elements: Macros and Functions *231*
7.4 Program Elements: Data Types, Data Structures, Modifiers, Statements, Loops and Pointers *233*
7.5 Use of Loops, Infinite Loops and Conditions *241*
7.6 Use of Function Calls *245*
7.7 Multiple Function Calls in Cyclic Order *246*
7.8 Function Pointers and Function Queues *247*
7.9 Queuing of Functions on Interrupts and Interrupt-Service-Routine Queues *248*
7.10 Embedded C and C++: Overview of Additional Features *250*
7.11 Objected-Oriented Programming *254*
7.12 Embedded Programming in C++ *254*
7.13 Optimisation of Codes and Memory Needs in Embedded C++ Programs to Eliminate the Disadvantages *255*
7.14 Embedded Programming in Java *256*
Summary *258*
Keywords *259*
Review Questions *262*
Practice Exercises *262*

8. Program Modeling Concepts **263-289**

8.1 Program Models *264*
8.2 Data-Flow Graph-Based Program Models *268*
8.3 State-Machine Programming Models for Event-Controlled Programs *272*
8.4 Modeling of Multiprocessor Systems *278*
8.5 UML Modeling *282*
Summary *287*
Keywords *287*
Review Questions *288*
Practice Exercises *288*

9. Real Time Operating Systems– I: Processes, Tasks and Threads and their Synchronization Using Inter-process Communication **290-331**

9.1 Multiple Processes in an Application *291*
9.2 Multiple Threads in an Application *293*
9.3 Tasks *294*
9.4 Task and Thread States *295*
9.5 Tasks and Data *297*
9.6 Clear-cut Distinction between Function, ISR, IST and Task by their Characteristics *299*
9.7 Inter-process Communication and Synchronisation *300*
9.8 Signals *301*

9.9 Concept of Semaphores *303*
9.10 Disabling and Enabling Functions *314*
9.11 Shared Data Problem *315*
9.12 Queues and Mailboxes *318*
9.13 Pipe and Socket Functions *323*
9.14 Remote Procedure Call Functions *328*
Summary *328*
Keywords *329*
Review Questions *330*
Practice Exercises *331*

10. Real Time Operating Systems II: Basic Functions of OS and RTOS 332-365

10.1 Operating System Services *333*
10.2 Process Management *335*
10.3 Timer Functions *335*
10.4 Event Functions *337*
10.5 Memory Management *338*
10.6 Device, File, and I/O Subsystems Management *339*
10.7 Interrupt Routines in RTOS Environment and Handling of Interrupt-Source Calls *344*
10.8 Introduction to Real-Time Operating Systems *347*
10.9 Basic Design Using a Real-Time Operating System *348*
10.10 RTOS Task-Scheduling Models *356*
10.11 OS Security Issues *357*
10.12 OS Standards: POSIX *358*
10.13 RTOS Interrupt Latency and Response Times of the Tasks as Performance Metrics *360*
10.14 OS Performance Guidelines *361*
10.15 Middleware: Meaning and Examples *361*
10.16 Application-layer Software: Meanings and Examples *362*
Summary *362*
Keywords *363*
Review Questions *364*
Practice Exercises *364*

11. Real-time Operating System Programming: MicroC/OS-II and VxWorks 366-424

11.1 Real-Time Operating Systems (RTOSes) *367*
11.2 µC/OS-II (MUCOS) *370*
11.3 Introduction to Unix-based Real-time Operating Systems *399*
11.4 RTOS VxWorks *400*
Summary *421*

Keywords 422
Review Questions 424
Practice Exercises 424

12. Real-Time Linux, Windows CE, OSEK, Handheld Devices and Automotives Operating Systems **425-460**

12.1 POSIX Compliant Operating Systems *426*
12.2 Real-Time Linux Operating systems *426*
12.3 Windows CE *436*
12.4 OSEK *451*
Summary 454
Keywords 456
Review Questions 459
Practice Exercises 459

13. Design Examples and Case Studies of Program-Modeling and Programming with RTOS **461-519**

13.1 Case Study of Coding for an Automatic Chocolate Vending Machine using MUCOS RTOS *462*
13.2 Case Study of Digital Camera *471*
13.3 Application to Communication—Network Router for IP Packets *477*
13.4 Embedded Systems in Robotics: Case Study of Orchestra Playing Robots *484*
13.5 RTOS for Control Systems *491*
13.6 Case Study of an Embedded System for an Adaptive Cruise Control (ACC) System in a Car *492*
13.7 General Language Characteristics, Features of MISRA-C for Embedded Programming in Automobiles *499*
13.8 Case Study of an Embedded System for a Smart Card, Access Control Systems (Smart Cards, RFIDs, Fingerscan) *500*
13.9 Case Study of a Mobile-Phone Software for Key Inputs *505*
Summary 514
Keywords 515
Review Questions 518
Practice Exercises 519

14. Embedded Software Development Process and Tools **520-546**

14.1 Introduction to Embedded Software-Development Process and Tools *520*
14.2 Host and Target Machines *525*
14.3 Linking and Locating Software *527*
14.4 Getting Embedded Software into the Target System *532*

14.5 Issues in Hardware Software Design and Co-Design *533*
14.6 Program-Level Performance Analysis and Performance Modeling *541*
14.7 Performance and Performance Accelerators *542*
Summary *543*
Keywords *543*
Review Questions *545*
Practice Exercises *546*

15. Testing, Simulation, and Debugging Techniques and Tools **547-564**

15.1 Integration and Testing of Embedded Hardware *548*
15.2 Testing Methods *550*
15.3 Debugging Techniques *553*
15.4 Laboratory Tools and Target Hardware Debugging *557*
Summary *563*
Keywords *563*
Review Questions *564*
Practice Exercises *564*

Appendix A *565*

Appendix B *567*

Index *573*