

# EMBEDDED SYSTEM DESIGN

# Embedded System Design

*by*

PETER MARWEDEL

*University of Dortmund, Germany*



A C.I.P. Catalogue record for this book is available from the Library of Congress.

ISBN-10 0-387-29237-3 (PB)  
ISBN-13 978-0-387-29237-3 (PB)  
ISBN-10 1-4020-7690-8 (HB)  
ISBN-13 978-1-4020-7690-9 (HB)  
ISBN-10 0-387-30087-2 ( e-book)  
ISBN-13 978-0-387-30087-0 (e-book)

---

Published by Springer,  
P.O. Box 17, 3300 AA Dordrecht, The Netherlands.

*www.springeronline.com*

*Printed on acid-free paper*

All Rights Reserved

© 2006 Springer

No part of this work may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, microfilming, recording or otherwise, without written permission from the Publisher, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work.

Printed in the Netherlands.

**This book is dedicated  
to my family.**

# Contents

Preface	xiii
Acknowledgments	xvii
1. INTRODUCTION	1
1.1 Terms and scope	1
1.2 Application areas	5
1.3 Growing importance of embedded systems	8
1.4 Structure of this book	9
2. SPECIFICATIONS	13
2.1 Requirements	13
2.2 Models of computation	16
2.3 StateCharts	18
2.3.1 Modeling of hierarchy	19
2.3.2 Timers	23
2.3.3 Edge labels and StateCharts semantics	24
2.3.4 Evaluation and extensions	26
2.4 General language characteristics	27
2.4.1 Synchronous and asynchronous languages	27
2.4.2 Process concepts	28
2.4.3 Synchronization and communication	28

2.4.4	Specifying timing	29
2.4.5	Using non-standard I/O devices	30
2.5	SDL	30
2.6	Petri nets	36
2.6.1	Introduction	36
2.6.2	Condition/event nets	40
2.6.3	Place/transition nets	40
2.6.4	Predicate/transition nets	42
2.6.5	Evaluation	44
2.7	Message Sequence Charts	44
2.8	UML	45
2.9	Process networks	50
2.9.1	Task graphs	50
2.9.2	Asynchronous message passing	53
2.9.3	Synchronous message passing	55
2.10	Java	58
2.11	VHDL	59
2.11.1	Introduction	59
2.11.2	Entities and architectures	60
2.11.3	Multi-valued logic and IEEE 1164	62
2.11.4	VHDL processes and simulation semantics	69
2.12	SystemC	73
2.13	Verilog and SystemVerilog	75
2.14	SpecC	76
2.15	Additional languages	77
2.16	Levels of hardware modeling	79
2.17	Language comparison	82
2.18	Dependability requirements	83

3. EMBEDDED SYSTEM HARDWARE	87
3.1 Introduction	87
3.2 Input	88
3.2.1 Sensors	88
3.2.2 Sample-and-hold circuits	90
3.2.3 A/D-converters	91
3.3 Communication	93
3.3.1 Requirements	94
3.3.2 Electrical robustness	95
3.3.3 Guaranteeing real-time behavior	96
3.3.4 Examples	97
3.4 Processing Units	98
3.4.1 Overview	98
3.4.2 Application-Specific Circuits (ASICs)	100
3.4.3 Processors	100
3.4.4 Reconfigurable Logic	115
3.5 Memories	118
3.6 Output	120
3.6.1 D/A-converters	121
3.6.2 Actuators	122
4. EMBEDDED OPERATING SYSTEMS, MIDDLEWARE, AND SCHEDULING	125
4.1 Prediction of execution times	126
4.2 Scheduling in real-time systems	127
4.2.1 Classification of scheduling algorithms	128
4.2.2 Aperiodic scheduling	131
4.2.3 Periodic scheduling	135
4.2.4 Resource access protocols	140
4.3 Embedded operating systems	143

4.3.1	General requirements	143
4.3.2	Real-time operating systems	144
4.4	Middleware	148
4.4.1	Real-time data bases	148
4.4.2	Access to remote objects	149
5.	IMPLEMENTING EMBEDDED SYSTEMS: HARDWARE/SOFTWARE CODESIGN	151
5.1	Task level concurrency management	153
5.2	High-level optimizations	157
5.2.1	Floating-point to fixed-point conversion	157
5.2.2	Simple loop transformations	159
5.2.3	Loop tiling/blocking	160
5.2.4	Loop splitting	163
5.2.5	Array folding	165
5.3	Hardware/software partitioning	167
5.3.1	Introduction	167
5.3.2	COOL	168
5.4	Compilers for embedded systems	177
5.4.1	Introduction	177
5.4.2	Energy-aware compilation	178
5.4.3	Compilation for digital signal processors	181
5.4.4	Compilation for multimedia processors	184
5.4.5	Compilation for VLIW processors	184
5.4.6	Compilation for network processors	185
5.4.7	Compiler generation, retargetable compilers and design space exploration	185
5.5	Voltage Scaling and Power Management	186
5.5.1	Dynamic Voltage Scaling	186
5.5.2	Dynamic power management (DPM)	189



<i>Contents</i>	xi
5.6 Actual design flows and tools	190
5.6.1 SpecC methodology	190
5.6.2 IMEC tool flow	191
5.6.3 The COSYMA design flow	194
5.6.4 Ptolemy II	195
5.6.5 The OCTOPUS design flow	196
6. VALIDATION	199
6.1 Introduction	199
6.2 Simulation	200
6.3 Rapid Prototyping and Emulation	201
6.4 Test	201
6.4.1 Scope	201
6.4.2 Design for testability	202
6.4.3 Self-test programs	205
6.5 Fault simulation	206
6.6 Fault injection	207
6.7 Risk- and dependability analysis	207
6.8 Formal Verification	209
References	212
About the author	227
List of Figures	229
Index	236

# Preface

## Importance of embedded systems

Embedded systems can be defined as information processing systems embedded into enclosing products such as cars, telecommunication or fabrication equipment. Such systems come with a large number of common characteristics, including real-time constraints, and dependability as well as efficiency requirements. Embedded system technology is essential for providing ubiquitous information, one of the key goals of modern information technology (IT).

Following the success of IT for office and workflow applications, embedded systems are considered to be **the** most important application area of information technology during the coming years. Due to this expectation, the term **post-PC era** was coined. This term denotes the fact that in the future, standard-PCs will be a less dominant kind of hardware. Processors and software will be used in much smaller systems and will in many cases even be invisible (this led to the term **the disappearing computer**). It is obvious that many technical products have to be technologically advanced to find customers' interest. Cars, cameras, TV sets, mobile phones etc. can hardly be sold any more unless they come with smart software. The number of processors in embedded systems already exceeds the number of processors in PCs, and this trend is expected to continue. According to forecasts, the size of embedded software will also increase at a large rate. Another kind of Moore's law was predicted: *For many products in the area of consumer electronics the amount of code is doubling every two years* [Vaandrager, 1998].

This importance of embedded systems is so far not well reflected in many of the current curricula. This book is intended as an aid for changing this situation. It provides the material for a first course on embedded systems, but can also be used by non-student readers.

## Audience for this book

This book intended for the following audience:

- Computer science, computer engineering and electrical engineering students who would like to specialize in embedded systems. The book should be appropriate for third year students who do have a basic knowledge of computer hardware and software. This book is intended to pave the way for more advanced topics that should be covered in a follow-up course.
- Engineers who have so far worked on systems hardware and who have to move more towards software of embedded systems. This book should provide enough background to understand the relevant technical publications.
- Professors designing a new curriculum for embedded systems.

## Curriculum integration of embedded systems

The book assumes a basic understanding in the following areas (see fig. 0.1):

- electrical networks at the high-school level (e.g. Kirchhoff's laws),
- operational amplifiers (optional),
- computer hardware, for example at the level of the introductory book by J.L. Hennessy and D.A. Patterson [Hennessy and Patterson, 1995],
- fundamental digital circuits such as gates and registers,
- computer programming,
- finite state machines,
- fundamental mathematical concepts such as tuples, integrals, and linear equations,
- algorithms (graph algorithms and optimization algorithms such as branch and bound),
- the concept of NP-completeness.

A key goal of this book is to provide an overview of embedded system design and to relate the most important topics in embedded system design to each other. It should help to motivate students and teachers to look at more details. While the book covers a number of topics in detail, others are covered only briefly. These brief sections have been included in order to put a number of

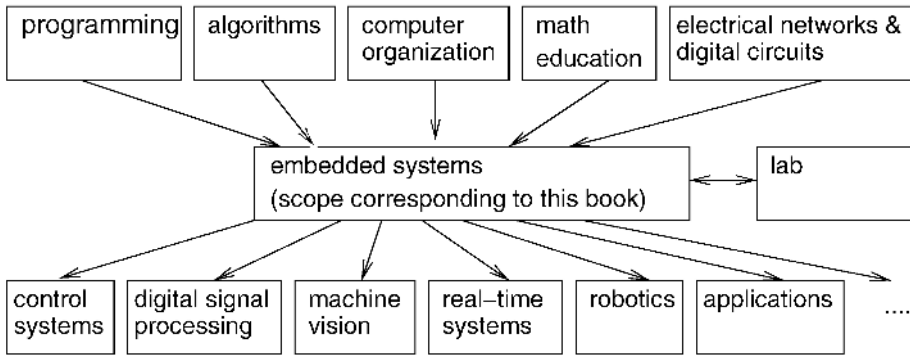


Figure 0.1. Positioning of the topics of this book

related issues into perspective. Furthermore, this approach allows lecturers to have appropriate links in the book for adding complementary material of their choice. The book should be complemented by follow-up courses providing a more specialized knowledge in some of the following areas:

- digital signal processing,
- robotics,
- machine vision,
- sensors and actors,
- real-time systems, real-time operating systems, and scheduling,
- control systems,
- specification languages for embedded systems,
- computer-aided design tools for application-specific hardware,
- formal verification of hardware systems,
- testing of hardware and software systems,
- performance evaluation of computer systems,
- low-power design techniques,
- security and dependability of computer systems,
- ubiquitous computing,
- application areas such as telecom, automotive, medical equipment, and smart homes,

- impact of embedded systems.

A course using this book should be complemented by an exiting lab, using, for example, small robots, such as Lego Mindstorm<sup>TM</sup> or similar robots. Another option is to let students gain some practical experience with StateCharts-based tools.

**Additional information related to the book can be obtained from the following web page:**

**<http://ls12-www.cs.uni-dortmund.de/~marwedel/kluwer-es-book>.**

This page includes links to slides, exercises, hints for running labs, references to selected recent publications and error corrections. Readers who discover errors or who would like to make comments on how to improve the book should send an e-mail to [peter.marwedel@udo.edu](mailto:peter.marwedel@udo.edu).

Assignments could also use the information in complementary books [Ganssle, 1992], [Ball, 1996], [Ball, 1998], [Barr, 1999], [Ganssle, 2000], [Wolf, 2001], [Buttazzo, 2002].

The use of names in this book without any reference to copyrights or trademark rights does not imply that these names are not protected by these.

Please enjoy reading the book!

Dortmund (Germany), September 2003

P. Marwedel

Welcome to the current updated version of this book! The merger of Kluwer and Springer publishers makes it possible to publish this version of the book less than two years after the initial 2003 version. In the current version, all typos and errors found in the original version have been corrected. Moreover, all Internet references have been checked and updated. Apart from these changes, the content of the book has not been modified. A list of the errors corrected is available at the web page listed above.

Please enjoy reading this updated book.

Dortmund (Germany), August 2005

P. Marwedel

## Acknowledgments

My PhD students, in particular Lars Wehmeyer, did an excellent job in proof-reading a preliminary version of this book. Also, the students attending my course “Introduction to Embedded Systems” of the summer of 2003 (in particular Lars Bensmann) provided valuable help. In addition, the following colleagues and students gave comments or hints which were incorporated into this book: W. Müller, F. Rammig (U. Paderborn), W. Rosenstiel (U. Tübingen), R. Dömer (UC Irvine), and W. Kluge (U. Kiel). Material from the following persons was used to prepare this book: G. C. Buttazzo, D. Gajski, R. Gupta, J. P. Hayes, H. Kopetz, R. Leupers, R. Niemann, W. Rosenstiel, and H. Takada. Corrections to the 2003 hardcopy version of the book were proposed by David Hec, Thomas Wiederkehr, and Thorsten Wilmer. Of course, the author is responsible for all remaining errors and mistakes.

Acknowledgments also go to all those who have patiently accepted the author’s additional workload during the writing of this book and his resulting reduced availability for professional as well as personal partners.

Finally, it should be mentioned that Kluwer Academic Publishers (now Springer) has supported the publication of the book from its initial conception. Their support has been stimulating during the work on this book.