Studies in Systems, Decision and Control

Volume 195

The series "Studies in Systems, Decision and Control" (SSDC) covers both new developments and advances, as well as the state of the art, in the various areas of broadly perceived systems, decision making and control–quickly, up to date and with a high quality. The intent is to cover the theory, applications, and perspectives on the state of the art and future developments relevant to systems, decision making, control, complex processes and related areas, as embedded in the fields of engineering, computer science, physics, economics, social and life sciences, as well as the paradigms and methodologies behind them. The series contains monographs, textbooks, lecture notes and edited volumes in systems, decision making and control spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution and exposure which enable both a wide and rapid dissemination of research output.

More information about this series at http://www.springer.com/series/13304

Alexander Barkalov · Larysa Titarenko ·
Małgorzata Mazurkiewicz

Foundations of Embedded Systems

Springer

Alexander Barkalov
Institute of Metrology, Electronics
and Computer Science
University of Zielona Gora
Zielona Gora, Poland

Małgorzata Mazurkiewicz
Institute of Control and Computation
Engineering
University of Zielona Gora
Zielona Gora, Poland

Larysa Titarenko
Institute of Metrology, Electronics
and Computer Science
University of Zielona Gora
Zielona Gora, Poland

Kharkiv National University of Radio
Electronics
Kharkiv, Ukraine

ISSN 2198-4182 ISSN 2198-4190 (electronic)
Studies in Systems, Decision and Control
ISBN 978-3-030-11960-7 ISBN 978-3-030-11961-4 (eBook)
https://doi.org/10.1007/978-3-030-11961-4

Library of Congress Control Number: 2018968085

© Springer Nature Switzerland AG 2019
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

In our daily lives, we meet with a huge number of computer systems. The civilized humanity is literally surrounded by various computer systems. Computers and embedded systems can be found in practically all fields of human activity. The first microprocessors appeared in the 1970s and gradually they became the basis of computer systems. Today, an overwhelming number of microprocessors are used in the implementation of various embedded systems. It is believed that embedded systems include more than 95% of all produced microprocessors. This trend will continue in the future. The development of embedded systems stimulates the development of the Internet of Things. In turn, this leads to an increase in the production of microprocessors. The circle closes, but this is not a vicious circle. Expanding the scope of use of embedded systems can significantly improve people's quality of life.

There are some specifics of embedded systems which distinguish them from universal computing systems. First of all, a particular embedded system executes a very limited amount of tasks. It is quite possible that only a single task is solving again and again. This allows a designer to optimize the hardware and software of a system. Second, embedded systems monitor and control some physical objects. To do it, they should interact with the physical world. This requires reconciling differences in the form of information representation and in the speed of information processing (cybernetic processes) and the speed of processes in a physical object. So, most of the embedded systems belong to the class of cyber-physical systems. Third, many embedded systems are mobile and require batteries as a source of electricity. So, for mobile and autonomous systems, it is especially important to reduce energy consumption. This extends their useful life. As a rule, universal computer systems do not have such problems. Their main task is to perform various calculations as quickly as possible. So, the designer of embedded systems must have specific knowledge.

The up-to-day state of the art in this area is characterized by three major factors. The first factor is a development of ultracomplex VLSI such as "system-on-programmable-chip" with billions of transistors and hundreds of millions of equivalent gates. The second factor is a development of hardware description languages such as VHDL and Verilog that permits to capture a design with tremendous complexness. The third factor is a wide application of different computer-aided design (CAD) tools to design very complex projects in the satisfactory time. These three factors affected significantly the process of embedded systems' design. Now the hardware design is very similar to the development of computer programs. There is no need to write embedded software using assembly languages. The mutual application of hardware description languages and CAD tools allows concentrating the designer's energy on the basic problems of design, whereas a routine work remains the prerogative of computers.

Tremendous achievements in the area of semiconductor electronics turn microelectronics into nanoelectronics. Actually, we observe a real technical boom connected with achievements in nanoelectronics. It results in development of very complex integrated circuits, particularly the different application specific integrated circuits and field-programmable gates arrays (FPGA). The largest of commercial chips have billions and billions of transistors. So, they are so huge that it is enough only a single chip to implement a very complex digital system such as a multi-processing system-on-a-chip. So, currently, there are practically all possibilities for implementing complex and diverse embedded systems.

We say "practically" because today some important factors are missing. First, there are not enough qualified designers to implement complex embedded systems. Second, there are no efficient software tools for formalized designing on both system and processor levels. Third, in Poland there are practically no monographs related to the design of embedded systems. If there are such books, they are dedicated to microcontrollers and microcontroller-oriented programming. However, embedded systems are not just microcontrollers. They may include programmable logic, intellectual property cores, bus arbiters, memory blocks, controllers of standard interfaces, and other blocks.

The book contains six Chapters and Conclusion. It is written by a research group from University of Zielona Góra (Poland). Professors Alexander Barkalov and Larysa Titarenko wrote Introduction, Chaps. 1–4, and Conclusion. Chapters 5 and 6 are written by Ph.D. Małgorzata Mazurkiewicz.

Chapter 1 provides some basic issues connected with embedded systems. Two types of computer systems are discussed (visible and invisible to a user). Next, some example is shown allowing understanding the basic features of embedded systems. The common features of embedded systems and their characteristics are discussed. The final section is devoted to basic requirements to embedded systems.

Chapter 2 is devoted to different design methodologies targeting complex embedded systems. The hierarchy of levels is introduced, as well as Y-Chart proposed by Prof. Daniel Gajski. Using the Y-Chart, some basic methodologies are discussed such as bottom-up, top-down, and meet-in-the-middle. The third section is devoted to platform-based design of embedded systems. Next, the peculiarities of ASIC-based design are discussed. The last section is devoted to very important problem of increasing the energy efficiency of embedded systems.

Chapter 3 presents some methods used for implementing computational algorithms in embedded systems. All methods are illustrated using the same end-to-end example. We start from the hardware implementation when some operational units (adders, multipliers, and other) are used to implement an algorithm during a single cycle of operation. Next, we discuss how to minimize hardware due to introducing an internal control unit. The third section is devoted to software implementation of computational algorithms based on microcontrollers. Some families of microcontrollers are analyzed. Next, we show how to choose blocks for implementing the embedded system for our example. We choose the microcontrollers ATmega32 to illustrate this implementation of the embedded system. At last, the heterogeneous implementation of computational algorithms is discussed. It is based on well-known conception of hardware-software co-design. We introduce the conception of hardware-software interface and show how to create an algorithm of its operation.

Chapter 4 deals with field-programmable gate arrays (FPGA). The basic stages are shown concerning evolution of programmable logic (from PROMs and PLAs to FPGAs). Next, the evolution of FPGAs is analyzed. Three ages of FPGAs are shown. Next, the modern FPGAs are produced by Xilinx and Intel (Altera). The last section is devoted to design methods targeting FPGAs. Basing on this information, we show how to execute calculations using digital signal processors, embedded memory blocks, and some components of Zynq 7000 by Xilinx.

Chapter 5 is devoted with FPGA-based implementing control algorithms represented using the language of graph-schemes of algorithms. We use models of Mealy and Moore finite state machines (FSM) to design the circuits of control units. We start from single-level FSM circuits implemented using look-up table (LUT) elements and/or embedded memory blocks (EMB). These methods are illustrated using standard benchmark FSMs. Next, the methods of structural decomposition are discussed. We discuss how to reduce hardware using the replacement of logical conditions, encoding the collections of microoperations, encoding the terms, and transformation of objects. We show how to use the pseudoequivalent states for optimization of logic circuits of Moore FSMs. The third section is devoted to hardware reduction of Moore FSMs targeting the replacement of logical conditions. At last, we discuss how to optimize hardware replacing state registers by state counters. The discussed methods target control units based on models of Moore FSMs.

Chapter 6 is devoted to programmable logic controllers (PLC). We start from the classification of PLC, their architecture, and cycle of operation. Next, the main laws of Boolean algebra are shown. We show the connection between the Boolean algebra and basic logic functions used in programming of PLC. Next, different programming languages used for PLC are shown. The last part is devoted to examples of programming for different Boolean functions and simple control algorithms. All programs are written using the Ladder Diagram language.

We understand that many questions have not been examined. This is almost impossible to do it in a single book. However, we have tried to provide basic information sufficient for self-study of other important problems related to embedded systems. We hope that our book will be interesting and useful for students and Ph.D. students in the area of Computer Science, as well as for designers of modern embedded systems.

Zielona Gora, Poland
October 2018

Alexander Barkalov
Larysa Titarenko
Małgorzata Mazurkiewicz

Contents

Abbreviations

ADC	Analog-to-digital converter
ALM	Adaptive logic module
ALU	Arithmetic and logical unit
ASIC	Application specific integrated circuit
ASSP	Application specific standard product
BE	Basic element
BF	Block of functions
BIMF	Block of input memory functions
BMO	Block of microoperations
CAD	Computer-aided design
CB	Communication block
CE	Communication element
CFM	Configuration flesh memory
CLB	Configurable logic block
CMO	Collection of microoperations
CMOS	Complimentary metal-oxide-semiconductor
CPLD	Complex programmable logic device
CPS	Cyber-physical system
CPU	Central processing element
DAC	Digital-to-analog converter
DC	Decoder
DSP	Digital signal processor
DST	Direct structure table
DVFS	Dynamic voltage and frequency scaling
EDA	Electronic design automation
EEPROM	Electrically erasable programmable read-only memory
EMB	Embedded memory block
EMBer	Logic block consisting from EMBs
ES	Embedded system
ESP	Embedded system platform

FF	Flip-flop
FPGA	Field-programmable gate arrays
FPLD	Field-programmable logic device
FSM	Finite state machine
GAL	Generic array logic
GFT	Generalized formula of transition
GOPS	Giga operations per second
GPU	Graphic processing unit
GSA	Graph-scheme of algorithm
HSC	Hardware-software co-design
HSI	Hardware-software interface
HW	Hardware
HWA	Hardware accelerator
I/O	Input-output
IC	Integrated circuit
ICU	Internal control unit
IE	Interface element
IMF	Input memory functions
IoT	Internet of things
IP	Intellectual property
ISA	Industry Standard Architecture
IT	Information technology
LAB	Logic array block
LAN	Local area network
LCD	Liquid-crystal display
LCS	Linear chain of states
LUT	Look-up table
LUTer	Logic block consisting from LUTs
MC	Microcontroller
ME	Memory element
MEMO	Mixed encoding of microoperations
MO	Microoperation
MoC	Model of computation
MP	Microprocessor
MPSoC	Multiprocessor system-on-a-chip
OS	Operating system
PAL	Programmable array logic
PC	Personal computer
PCB	Printed circuit board
PCI	Peripheral component interconnect
PE	Processing element
PES	Pseudoequivalent states
PG	Productivity gap
PI	Programmable interconnect
PLA	Programmable logic arrays

PLL	Phase-locked loop
PLS	Programmable logic sequencer
PMM	Power-mode management
PROM	Programmable read-only memory
PU	Processing unit
RAM	Random access memory
RG	Register
RLC	Replacement of logical conditions
ROM	Read-only memory
RTL	Register transfer level
RTOS	Real-time operating system
RTS	Real-time system
SB	Switching block
SBF	System of Boolean functions
SFT	System of formulae of transitions
SG	System gate
SoC	System-on-a-chip
SOP	Sum of products
SPLD	Simple programmable logic device
SRAM	Static random access memory
SSI	Stacked silicon interconnection
STG	State transition graph
STT	State transition table
SW	Software
TOC	Transformation of object codes
UFM	User flash memory
ULSI	Ultra large scale integration circuit
USB	Universal serial bus
VGA	Video graphic array
VLSI	Very large scale integration circuit