Microcomputer Interfacing

G. Jack Lipovski

LexingtonBooks

Microcomputer Interfacing

Principles and Practices

G. Jack Lipovski
University of Texas

LexingtonBooks
D.C. Heath and Company
Lexington, Massachusetts
Toronto

Library of Congress Cataloging in Publication Data

Lipovski, G. Jack
Microcomputer interfacing.

1. Microcomputers. 2. Computer interfaces. I. Title.
TK7888.3.L49 001.64′404 79-9683
ISBN 0-669-03619-6

Fourth printing, August 1982

Published simultaneously in Canada

Printed in the United States of America

International Standard Book Number: 0-669-03619-6

Library of Congress Catalog Card Number: 79-9683

Microcomputer Interfacing

Lexington Books Series in Computer Science

Kenneth J. Thurber, General Editor

Basic Computer Logic
John R. Scott

Comparative Studies in Software Acquisition
Steven Glaseman

Computing in the Humanities
Peter C. Patton and *Renee A. Holoien*

Data Base Computers
Olin H. Bray and *Harvey A. Freeman*

Database Management System Anatomy
James A. Larson

Data Structures and Computer Architecture
Kenneth J. Thurber and *Peter C. Patton*

Distributed Database Management Systems
Olin H. Bray

Distributed-Processor Communications Architecture
Kenneth J. Thurber and *G.M. Masson*

Interconnection Networks for Large-Scale Parallel Processing
Howard J. Siegel

Local Networks
W.R. Franta and *Imrich Chlamtac*

Microcomputer Interfacing
G. Jack Lipovski

Requirements-Oriented Computer System Design
Kenneth J. Thurber

Software Blueprint and Examples
Yaohan Chu

User-Designed Computing
Louis Schlueter, Jr.

Dedicated to my wonderful wife

Belle

Preface

The microcomputer has changed so many aspects of engineering that the cliché "microcomputer revolution" appears in numerous magazine articles and learned papers in the field. It is a great tool. Microcomputer-systems designers, however, have had to develop their own foundations, based on trade magazine articles and applications notes, because no available textbooks or reference books have specifically developed those foundations. This book aims to teach the principles and practices of microcomputer-systems design in general and interfacing in particular.

This book was developed from a set of notes for a senior-level course in microcomputer design. The course focuses on the combined hardware-software design of microcomputer systems. It emphasizes the principles—the theory—of design because that is as necessary for a solid foundation for system design as is theory in any engineering discipline. Even so, it also emphasizes the practices—the details of how to get a system to work—because microcomputer-system design requires hands-on experience. The difference between a student who merely reads about microcomputers and a student who has worked with one is clear evidence that theory must be taught with practice. Practical experience is desirable in almost any engineering course. This is not always possible. But microcomputer systems are inexpensive enough that the school or the student can afford this hands-on opportunity, and the joy of seeing the principles work is so attractive that the student often cannot get enough of the material. So the course and this book emphasize both the principles and the practices of microcomputer design.

These principles and practices must cover both hardware and software. A purely hardware approach might be more attractive to a seasoned digital-system designer or an electrical engineering student, but that approach leads to ignorant choices that either do not take full advantage of the tremendous power of software or that force unnecessary constraints and therefore higher costs on its development. On the other hand, a purely software approach misses the opportunity to understand how and why things happen and how to take advantage of these things.

A combined hardware-software approach does require more background. The course on which this book is based is the second course of a two-course sequence. The first course teaches basic assembler-language programming. It is really just a standard computer-science course on assembler-language programming, but it is taught with an assembler language of a microcomputer. The second course builds upon that background. It also presumes some background in logic design, such as would be obtained in a standard introductory course on that topic. This book, however, has two chapters that survey the two prerequisite courses. These two chapters can be skimmed as a review of the required background material, or they can be carefully studied as a condensed tutorial if the reader has not had the earlier courses or their equivalent material. The two chapters are comparatively compressed and terse, because they are intended as review or intensive tutorial material preliminary to the main subject of the book, which follows in the later chapters.

The practices discussed in this book are made concrete through detailed discussion of the Motorola M6800 family of microcomputers and the M6809 microprocessor instruction set. These are used as a means to the end of teaching principles and practices in general, rather than for the promotion, sale, or use of Motorola products. Applications, notes, and catalogs are available from Motorola to that end. Specific and detailed discussion encourages and aides the reader in learning through hands-on experience, and it vitally contributes to an enthusiasm for and understanding of the principles. The M6800 family is used primarily because the M6809 is believed to be the most easily taught microcomputer. Its instruction set is as complete as that of any other machine, supporting enough addressing modes and index registers to teach the intelligent use of data structures, and it is symmetrical and comparatively free from quirks and anomalies that detract from the subject under discussion. Nevertheless, detailed comparisons between the M6809 and other well-designed microcomputers clearly show that others may be better than the M6809 for different applications. We do not mean to imply by the selection of the M6809 that it is the best microcomputer for all applications. We want to stress this point to the reader; however, a comparative study of different microcomputers and applications is beyond the scope of this book. On the other hand, the M6800 family, and the M6809 microcomputer in particular, are excellent vehicles for teaching the principles of microcomputer design.

The first two chapters quickly survey the background needed for the remainder of the book. Chapter 1 covers the instruction set of the M6809. It is intended as a survey for a reader who is acquainted with some other assembler language, either of another microcomputer or of a large machine. Chapter 2 covers basic computer organization but confines itself to those aspects of the topic that are particularly germane to microcomputers. For example, basic computer organization traditionally covers floating-point arithmetic, but this chapter does not; and this chapter dwells on tristate buses, although these are not often covered in discussions of computer organization. The rest of the book covers three basic themes: parallel input/output (I/O) ports in chapters 3 and 4, analog components in chapters 5 and 6, and communications devices in chapter 7. The simple parallel I/O port is displayed in chapter 3. Hardware and software aspects are studied side by side, so that the reader need no longer be intimidated by the basic interface. Chapter 4 discusses interrupts and their alternatives. Hardware-software tradeoffs are analyzed, and different techniques are exemplified. Chapter 5 surveys the traditional (voltage) analog components that are commonly used in microcomputer I/O systems. Sufficient information is provided so that the reader can understand the uses, the limitations, and the advantages of analog components and can use this chapter as a springboard to other texts, magazine articles, or discussions with colleagues with a fuller understanding of analog design. Chapter 6 introduces the counter-timer as an interface to frequency-analog signals. Techniques to generate signals with a frequency, or to measure a signal with a frequency, that is analog to some quantity are discussed. Moreover, the hardware-software alternatives to using this most interesting integrated circuit are discussed and evaluated. Chapter 7 describes communications devices. The Universal Asynchronous Receiver Transmitter (UART) and related devices are thoroughly studied, and other communications techniques are described.

This book is intended to be complete in itself, with no additional books necessary. If the reader wishes to skip the problems at the end of each chapter, and to omit doing the laboratory experiments suggested in the text, this book is sufficient. However, we recommend doing the problems and some laboratory experiments, and to do so the reader will need a copy of *The Complete Motorola Microcomputer Data Library*, printed after 1 June 1980. (Earlier printings have an incomplete description of the 6809.) Most readers will either have that book or will want to acquire a copy for further information on the chips that can be interfaced to the M6809. However, our two appendixes on an Intersil UART and a Motorola addressable UART contain material that is not in the *Data Library* and will allow the reader to do all the problems. The reader may also want to refer to *The TTL Data Book for Design Engineers* published by Texas Instruments, Inc. Also, we are planning a laboratory manual (which we hope will be available by December 1980) and a solutions manual, which may be obtained by contacting the author.

Some remarks are necessary on the style of this book. Terms are formally introduced and used as carefully as possible, and an extensive index is provided to help locate explanations of these terms. This is necessary in order to make some sense out of a subject as broad and rich as microcomputer-system design. There is no room for muddy terminology or the use of undefined jargon. Even though the topic is under rapid development and the terminology used in trade journals and manufacturers' applications notes is inconsistent and often contradictory, the terminology used in a text must be clear and consistent. However, a book full of definitions is too hard to read. The first version of the course notes that led to this book tended to be ponderous. Moreover, today's students are more attuned to television colloquialism and are tuned out to ''third-person boring,'' which is often the style of learned textbooks. So we condescend to first-person conversational, and we enjoy it. The ''we'' in this book stands not only for the author but also for his colleagues, his teachers, and his students, who have taught him a great deal and who have collectively inspired and developed the principles and practices we discuss in the book. But we admit to exploring Webster's *Collegiate* occasionally for just the right word because we enjoy the challenge, and we even allow a pun or two where it does not interfere with the presentation of the material. We cannot deny it—microcomputer design is fun, and we have fun talking about it. Please forgive us if it shows in the style of the book.

Acknowledgments

The author would like to express his deepest gratitude to all the people who have contributed to this book. In addition to a number of students, fellow faculty members, and colleagues in industry who have helped in no small part, special thanks are due to Professor Charles Roth, who taught the course using these notes and who corrected many errors and contributed some of the better problems at the end of each chapter. Deep thanks are given Terry Ritter at Motorola, one of the designers of the M6809, for his extensive review of the manuscript, and for the many corrections and suggestions that make the book so much better. I would like to thank Lance Leventhal and Harold Stone for some fine comments in their reviews of the manuscript; these comments were invaluable. Gratitude is also offered to Professor Terry Wagner and students Ed Upchurch and Jeff Oliver, who proofread some of the chapters. Finally, I acknowledge the support of my secretary, Janelle Dolby, for correcting errors on the text edition, and my technician, Pat Horne, for preparing some of the rough-draft figures. If I omitted anyone, may I just say that throughout the book, the word *we* refers to me and to all of you who have helped in the formation of the ideas and in the generation of this book.

Contents

List of Figures

List of Tables