Programming Microprocessor Interfaces for Control and Instrumentation

MICHAEL ANDREWS

Programming Microprocessor Interfaces for Control and Instrumentation

MICHAEL ANDREWS

Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging in Publication Data

Andrews, Michael
Programming microprocessor interfaces for control and instrumentation.

Bibliography: p.
Includes index.
1. Microprocessors—Programming.
2. Computer interfaces. I. Title.
QA76.6.A54 001.64'04 81-5932
ISBN 0-13-729996-6 AACR2

Editorial/production supervision and interior design by Mary Carnis
Cover Design by Mario Piazza
Manufacturing Buyer: Joyce Levatino

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Prentice-Hall International, Inc., *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall of Canada, Ltd., *Toronto*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Prentice-Hall of Southeast Asia Pte. Ltd., *Singapore*
Whitehall Books Limited, *Wellington, New Zealand*

Preface

Microprocessors are a truly versatile design tool. This new dimension literally expands the horizon of applications beyond imagination. Yet to capture the power of microprocessors, you must understand both the hardware and the software aspects of these machines. Unlike any other computer technology, the microprocessor is available to everyone because of its low cost. Unfortunately, few recognize the hidden costs of software development. As in large computers, we can be plagued by code that is illogically structured, very application dependent, and totally undocumented. The new generation of microprocessor architectures and software architectures leaves us no room for writing bad code. It is my hope that you will come away from this book with a clearer grasp of modern programming methods, which, when consistently applied, minimize frustration, errors, and programs which are expensive to maintain.

This book is written for the technician, engineer, or programmer who must design or employ a microprocessor in the many applications of signal processing and control. Nine chapters carry you through the basics of a popular 8-bit microprocessor, illustrating each point with examples and programs. The book also serves as a technical reference for the designer of microprocessor-based equipments. Only a modest electronics background is necessary to understand the easy interfacing principles. Hence, practicing engineers, scientists, and programmers will find this book useful in their applications. The many examples and exercises at the end of each chapter should help the instructor using this book as a class text.

The book is divided into two parts. The first half develops the hardware and software architecture of a microprocessor system centered about the 6809 microprocessor unit. The second half first develops software through modern programming techniques. Later chapters describe many practical applications for microprocessors. Chapter 1 establishes themes for each remaining chapter, with each section serving as the entry point to topics more fully developed later in the text. This chapter is your road map to further study throughout the text. Chapter 2 introduces the software and hardware architecture of the popular 6809 microprocessor. Here I present the essential features of each architecture, which will help you to understand the power of this device. Only the important factors and properties crucial to the design and implementation of microprocessor units are included, thus enabling you to incorporate the microprocessor unit in your design as quickly as possible. Later in the chapter, the important notions of position independence and structured programming are developed.

Interfacing a microprocessor to the real world is no small effort. Most devices have numerous control and timing signals, which must be clearly understood before any interface can be developed. In Chapter 2 I also analyze the 6809 signal set and electrical characteristics. My intent is to introduce important control signals which lead to better designs. Here you will find discussion on the address, data, and control buses. In all microprocessor systems, there are master synchronization signals, from which all peripherals are clocked in step. In this chapter I explain such use of the E and Q signal functions in the 6809.

In Chapter 3 I discuss procedures for configuring a microprocessor system, related to bus loading and expansion. Finally, I analyze the all-important memory interface techniques between ROMs, RAMs, and the MPU.

The true personality of a microprocessor is its instruction set. In Chapter 4 we study several of the important instructions in a microprocessor taken from four classes: *data movement*, *data manipulation*, *program movement*, and *program status*. Here we see how registers, memory locations, and stacks are employed in microprocessor programs. My goal is to develop your understanding of data and instruction flow to the microprocessor unit through the various buses. At the same time, you will learn how cycle times in a microprocessor are used. In the last half of this chapter I describe assembler and editor features that support the 6809 code development.

Chapter 5 is the most important chapter. By example, I show you how code should be written. This chapter on modern programming methods consistently employs modular programming techniques, develops your understanding of position independency, and describes recursive and reentrant programs. These programs stress structure and position independence because I have seen the unbearable cost of generating code without these attributes. There is literally no excuse for developing code that is not modular and structured. By employing such techniques, you can reap many benefits from code that is easily maintainable and debuggable, resilient to misapplication, readily understandable, and clearly documentable. In this chapter I have included many useful programs necessary for such vital tasks as floating-point arithmetic, multibyte multiplication and division routines, and text string searches. Because most applications require fast execution in minimal storage, a good solution is to program at the assembly language level. For these reasons, in this text I have focused on programming topics at the machine language level. Even so, the techniques so described apply equally well to high-level language development. Should you enjoy the luxury of a high-level language such as BASIC, FORTRAN, PASCAL, or PL1, the material should be beneficial.

Chapter 6 is our turning point from the software architecture dimension to the interface design dimension in a microprocessor system. Here I develop simple single-wire interfaces which lay the foundation for parallel, serial, and analog interfaces. All of these interfaces are described in this chapter with numerous examples using a combination of software and hardware approaches. In this chapter you will find interfaces with the 6821 PIA and the 6850 ACIA.

In Chapter 7 we complement the hardware focus on interfaces with the software requirements necessary for input/output programming. All important topics as real-time programming, interrupts, and interrupt-driven systems are described. From this chapter you will gain considerable insight into the need for counting cycle time and the number of program bytes in memory.

The microprocessor plays a pivotal role in the design and implementation of data acquisition and control systems. Chapters 8 and 9 present essential topics in signal sampling and conversion. In Chapter 8 I discuss transducers, standard industry functions for instrumentation, determination of aperture time for sampling analog signals, and popular codes for A/D and D/A conversion. My goal is to make you aware of design requirements for the front end of microprocessors, including low-pass filtering, noise reduction, and signal averaging. In Chapter 9 several important digital control algorithms for the 6809 with actual programs are presented. A useful development technique for translating mathematical descriptions of a control system from its Laplace model to the difference-equation form, and finally to its digital control implementation suitable for programming, are also described. The useful control topics include proportional–integral–derivative control and deadtime compensation. As with all chapters, a number of examples with actual microprocessor programs is offered.

I have chosen the 6809 architecture for a variety of reasons. First, the 6800 family of microprocessors is very popular. Second, its instruction sets closely resemble those found in many larger computers, thus enabling the minicomputer and maxicomputer user to rapidly grasp concepts in the microprocessor world. Third, the instruction set of the 6809 is powerful, permitting the invocation of modern programming methodology. Its instruction consistency, versatility, and flexibility will help you to generate quality code. Even though I have focused upon the 6809, much of the material in this text can be applied to other architectures and microprocessor systems. However, I have found in practice that demonstrating the power of one real machine is worth far more than generalizing over abstractions that may have no practical relevance.

I gratefully acknowledge contributions from Lothar Stern, Tim Ahrens, and Bob Burlingame of Motorola Semiconductor Products, Inc., for technical assistance and for permission to reprint from several fine data specification bulletins. In addition, Lynn Schimanuki, Bruce McGreggor, Jonathan Dust, and Vish Dixit contributed much to the generation of several programs. I appreciate the support of Terry Ritter and Lloyd Maul of Motorola for many stimulating discussions and contributions to this text. Without their guidance, this effort might well have fallen short of its desired objectives. Finally, I acknowledge the typing support of Todd Gale, a most enthusiastic person and my wife, Sandra, for her loving devotion to me.

Michael Andrews

Contents

1

Introduction to the Microprocessor World

1.1 WHERE DO WE FIND MICROPROCESSORS?

Everywhere! Look at your wrist. In the kitchen. What awoke you up this morning? Do you know how the tuner works on your television? How did you compute your monthly budget? In countless situations today, we enjoy the power, flexibility, and convenience of microprocessors. Following are just a few of the recent applications of microprocessors:

- Microwave ovens
- Digital watches
- Digital clocks
- Smart oscilloscopes
- Intelligent terminals
- Small computers
- Data-acquisition modules
- Patient-monitor systems
- Energy-management systems
- Process controllers
- Modems
- TV games
- Calculators
- Telephones
- Educational toys
- Radios
- CB scanners
- Home computers
- Gas chromatographs
- Telecommunications
- TV tuners

Process Control

The list of applications illustrates two significant points. First, microcomputers (microprocessors with other devices to make a "system") are not only replacing minicomputers in some applications, but more important, microprocessors are creating many new market areas. Second, microprocessors are a truly versatile design tool. The low cost, small size, and lower power consumption of microprocessors increasingly convinced designers to utilize them in a wide range of applications. Furthermore, the growing tendency among users is to desire local control by microcomputer application rather than to employ large centrally located computers remotely controlled. Many separate locations create an ever-widening demand for microprocessors, especially in distributed microprocessing. Local operators can control, monitor, and visually oversee the actual effect of the microprocessor's operation instantly, with little inherent delay. Microprocessors increasingly replace large centrally located mini- or maxi-computers previously used in process control applications.

The use of microprocessors as local microcontrollers at various stages of a process has several advantages. They provide convenient access by local line personnel to correct problems and fine-tune a system. Miles of expensive cabling are eliminated, costs go down, and communication between local operators and remote operations is reduced, thereby also reducing the possibility of communication errors. The trend to return control to the actual process site through the use of local microcontrollers is thus very strongly based.

In process control applications, we typically find hundreds of measurement points, such as temperature, pressure, line speed, and other process data. In operation, numerous analog circuits can be automatically selected and digitized under the control of a microprocessor. The microprocessor performs the computation and processing operations at lightning speeds, switching inputs and digitizing the analog data. It is not uncommon to find a microprocessor also generating tens of analog signals via digital-to-analog converters driving actuators in manufacturing processes to complete the real-time computer-aided process loop. Microprocessors can also send the return data to local monitoring stations. They can process status and display it on bar graphs via a color cathode ray tube (CRT) while simultaneously displaying several process variables on digital meters.

Instrument Panels

Microprocessors are a natural choice for smart instruments. They conveniently scan the many pushbuttons on the user's panel, monitor input signals, and generate digitized output signals—all while annunciating data on light-emitting diodes (LEDs). The digital voltmeter in Figure 1.1 and the synthesizer/function generator in Figure 1.2 are typical of the microprocessor implementation in modern instrumentation. Microprocessors help