Microcomputer Interfacing

HAROLD S. STONE

Microcomputer Interfacing

HAROLD S. STONE
UNIVERSITY OF MASSACHUSETTS, AMHERST

WESLEY PUBLISHING COMPANY
READING, MASSACHUSETTS
MENLO PARK, CALIFORNIA
LONDON
AMSTERDAM
DON MILLS, ONTARIO
SYDNEY

This book is in the **ADDISON-WESLEY SERIES IN ELECTRIC ENGINEERING**

SPONSORING EDITOR: *Tom Robbins*
PRODUCTION EDITOR: *Marilee Sorotskin*
TEXT DESIGNER: *Herb Caswell*
ILLUSTRATOR: *Jay's Publishers Service Inc.*
COVER DESIGN AND ILLUSTRATOR: *T. A. Philbrook*
ART COORDINATOR: *Joseph Vetere*
PRODUCTION MANAGER: *Sue Zorn*
PRODUCTION COORDINATOR: *Helen Wythe*

The text of this book was composed in Times Roman on a Mergenthaler 202 by Information Sciences Corporation and was printed by R. R. Donnelley and Sons.

Library of Congress Cataloging in Publication Data

Stone, Harold S., 1938–
Microcomputing interfacing.

Bibliography: p.
1. Interface circuits. 2. Microcomputers—Circuits.
I. Title. TK7868.158S76 621.3819'5[illegible] 81-17619
ISBN 0-201-07403-6 AACR2

ISBN 0-201-07403-6
ABCDEFGHIJK-DO-898765432

INSTRUCTOR'S PREFACE

In decades to come historians will look back to the 70s and 80s as the era of the Computer Revolution. Just as the Industrial Revolution marks the period when people learned to harness energy to drive machinery, so the Computer Revolution marks the period when people have learned to harness microelectronics for control and for information processing. Within of a decade of its introduction, the microprocessor made an enormous impact on the way we live and work. Who can recall the annoying difficulty of doing long division by hand, to say nothing of such transcendental calculations as compound interest, square roots, and trigonometry? Today's electronic calculator places all of these calculations at your fingertips at essentially zero cost. But the calculators of the early 70s were only precursors of the innovations to come.

The microprocessor has spawned the personal computer, the small-business computer, the word processor, the data communications network, the automotive computer, and the intelligent telephone branch exchange. Yet this is still only a beginning. The microprocessor has been used in instruments and appliances in place of mechanical or discrete electric components. In such applications it yields lower cost, greater reliability, and greater functionality than former designs. But the microprocessor also makes possible a new technology that heretofore has not existed. In the laboratory, the logic analyzer is a microprocessor-controlled oscilloscope that enables the engineer to capture and analyze electronic waveforms in a manner that had never been possible before. In medicine, the microprocessor has led to the CAT-scanner and the ultrasonic scanner that have dramatically improved the ability to diagnose illnesses. In merchandising, the point-of-sales terminal that reads product labels automatically provides greater control of inventory, as well as faster, more accurate handling of sales. The microprocessor-controlled reading machine that automatically scans printed text and speaks the words it sees aloud has brought new vision to the blind. The main limitation on the innovative applications of microprocessors today is not technological, but rather one of imagination and skill.

To help surmount the human limitation, this textbook is offered as one way to introduce the student to basic principles of microprocessor technology. A primary goal is to increase the pool of innovators. It is oriented to an undergraduate curriculum, and is ideally suited for juniors and seniors in a Computer Engineering, Electrical Engineering, or Computer Science program. A course based on this text will give the student a strong foundation in techniques for connecting computers to peripherals and communications devices, and in the methodology for programming the computer to control external devices in real time.

The well-prepared reader has had instruction in hardware, software, electronics, and mathematics. But the material presented here is modular so that an instructor can skip portions of the text that refer to topics that students at particular institutions will not yet have mastered. A breakdown of the prerequisites is as follows:

1. *All Chapters*: The student should have some exposure to microprocessors and logic design through such textbooks as Blakeslee (1979), Klingman (1977), Kraft and Toy (1979), Krutz (1980), or Peatman (1977). This general type of textbook introduces the student to the logic components, design techniques, and the structure of digital computers. The student should be comfortable with assembly language, but need not have extensive skills in this area. It is desirable for the student to have read such textbooks as Gear (1980) or Wakerly (1981) that cover assembly language for many different machines.
2. *Chapter 2, Transmission Lines*. Prior exposure to transmission-line theory is helpful for Chapter 2, but not absolutely necessary. The chapter is self-contained in that all background required to support the physical concepts is developed within the chapter. In curricula in which the electronic aspects of interfacing have been omitted, the instructor may choose to skip portions of Chapter 2 (grounding, shielding, and transmission lines), Chapter 3 (bus interconnections), and Chapter 7 (magnetic-recording techniques).
3. *Chapter 7, Linear Systems*. Some results and equations in this chapter are cited from other sources rather than derived in the chapter. A student should know Laplace transforms and transfer functions for full appreciation of this material. The exercises require a knowledge of electronics as well. The material may be skipped at institutions that teach microprocessor interfacing earlier than linear systems.
4. *Chapter 9, High-Level Language Programming*. The student should be familiar with some high-level language such as Pascal, ALGOL, Ada, PL-I, FORTRAN, or COBOL. The notation in the chapter is basically Pascal, but should be quite understandable for readers familiar with any of the first four languages cited here. Where the only high-level language in the curriculum is FORTRAN or COBOL, before the students read Chapter 9, the instructor may wish to incorporate a brief tutorial on a block-structured language.

The presentation of material in the text is three-tiered. Each chapter contains

1. Basic principles.
2. Applications of the principles in present technology.
3. Specific examples of the use of the principles.

Principles are stressed by necessity. Principles tend to be the foundation of expertise. They tend to change very slowly, if at all, over long periods of time. Details and specific facts quickly become obsolete. In the microprocessor industry, new generations of memory and microprocessors appear every two to three years. This means that details taught to a sophomore will be obsolete by the time that sophomore graduates. A curriculum must, therefore, rest on the principles that support the technology. The student must master these first, and must learn to apply them. As the industry advances and the specific details change, the student must be able to adapt to these new details without outside instruction. Therefore, a college curriculum should prepare the student for self-education in the future. To do so requires a thorough foundation in basic principles.

The nine chapters in this text are more than sufficient for a semester course in microprocessor interfacing. The instructor can easily select a subset of material to adapt the text to any particular curriculum. Core material that should be in all curricula consists of

Chapter 1, basic microcomputer structure
Chapter 3, bus interconnections
Chapter 5, serial interfacing
Chapter 6, parallel interfacing
Chapter 9, software development

Curricula in which electronic design and logic design is stressed should add

Chapter 2, grounding, shielding, and transmission-line techniques
Chapter 4, memories

Curricula that stress the use of the microcomputer as a control element should incorporate

Chapter 7, magnetic-recording techniques
Chapter 8, CRT-controller design

Now let's turn to methods of instruction for the material. Lectures should be coupled with a computer laboratory in which the student can perform simple interfacing experiments including the development of elementary control software. The experimental laboratory in conjunction with the course should occupy roughly three hours per week and should be followed later in the curriculum by one or more project-design laboratories devoted to microprocessor-based designs. The project-design laboratory gives the student an opportunity to integrate information from many subject areas, such as interfacing techniques, software development, and communications.

The experiments in the text are sufficient for a full semester of laboratory work. The reason for an experimental orientation instead of a design orientation is that the information is passed quickly and efficiently when the student does not have to design and debug the bulk of the experimental apparatus. The student uses existing equipment and commercially available boards to learn the principles of interfacing. The student exercises the equipment through small digital project boards and simple interfacing software. By observing the behavior of the equipment on oscilloscopes and logic analyzers, the student learns about such basic notions as noise reduction, electrical loading, timing, hysteresis, handshaking, skew, etc. The experiments in a laboratory should demonstrate various phenomena and should illustrate preferred approaches for dealing with fundamental problems. After completing the experimental lab, the student should be well prepared for subsequent project-design labs.

It would be rather ironic in this age of high technology to approach microprocessor education in a totally traditional form. This textbook is an example of a technology that is centuries old. Obviously, the printed word is an effective way for presenting information because it would not have survived to this day if it were not. But can we do better? In part ticular instances, new technology enables one person to do in one day what formerly took four people to do in a week. Can new technology help the academic community educate students more effectively and efficiently? The search for a better way has led this author to develop an instructional system for this course consisting of this text plus a set of color video tapes. Tapes are produced by the Association for Media-Based Education for Engineers (AMCEE) at Georgia Institute of Technology in Atlanta, Georgia. The tapes may be ordered by writing to Addison-Wesley Publishing Company, Reading, MA 01867; Attention: Tom Robbins, Acquisitions Editor, Computer Engineering.

The instructional system is modular in that the text stands on its own and can be used in the traditional ways. The tapes too are self-contained and can be used independently. Together, the tapes and the text make up an instructional system that is far more effective than either medium by itself.

To see how the two work together, consider the material on transmission lines in Chapter 2. The student is told about reflections on transmission lines and how terminations can remove or reduce reflections. The student has to see this to appreciate the ideas fully. Experiments illustrating the ideas are demonstrated on the video tapes. Because the experiments show waveforms changing in time, and show them with their normal spikes and jitter rather than as idealized waveforms, the student gains experience with the real world rather than an artificial one. The key here is that there is information in the dynamics of the video image as it changes in time. That information is lost when the image is photographed or drafted as a figure in a textbook. When specific comparisons are made on the videotape, the images appear in rapid succession so that the student can quickly grasp what similarities and differences exist. The behavior of a phase-locked loop acquiring phase lock appears vividly on video as a sudden change of frequency of a voltage-controlled oscillator. The jitter in the oscillator at the threshold of acquisition is clearly visible. How can this information be displayed in a textbook? For waveforms changing in time, the video image is clearly superior to the printed image.

This author has often prepared classroom or laboratory demonstrations to illustrate basic ideas. The effort involved in setting up a demonstration is considerable and not always successful. Some demonstrations work well on the bench, but fail when the equipment is moved to the classroom. A probe may fall off, or a connection might not be tight, or a noise "glitch" may cause the logic to latch in a failure mode. The experiments on the video tape all work correctly. They were carefully set up and video taped in operation so that the course instructor need not repeat the effort in the setup nor take the risk of the experiment failing.

The video medium leads to a better presentation of the waveforms than does the actual physical equipment. The physical size of an oscilloscope or logic-analyzer display is only 10 to 20 cm square, which is too small for a classroom of 30 students. This forces the instructor to set up the experiments in a laboratory and demonstrate them to small groups of students so that each has an opportunity to study the principles being demonstrated. Apart from the inefficiency of this method, it does not resolve the basic difficulty of pointing to specific places on a small display screen. Pointing at the images is not very effective because the instructor's hand tends to block large portions of the display from view. However, the image of the same display on the video tapes is superior to the oscilloscope because the image is enlarged. Moreover, electronic superposition techniques permit the instructor to point to and label the most highly detailed parts of a waveform with no obstruction of view to other parts of the image. The video tapes relate the waveform to a schematic by using zoom and pan to illustrate various regions of the schematic, with those images juxtaposed between images of waveforms that appear at selected points on the schematic. The information is presented at an extraordinarily fast pace compared with classroom discussion, yet is easily comprehensible because of the way the video medium is used to advantage. Should the student wish to review specific waveforms, it is a very simple matter to rewind to the point of interest and play the material again. Consequently, the video tapes are an extremely attractive solution to the laboratory demonstration problem. There is no setup overhead, the equipment is inexpensive, and video is more effective than the laboratory equipment itself for reaching large groups of people. The tapes, like the text, are prepared in a modular fashion so that the instructor can select specific material to support lectures and laboratory work.

Because of rapid changes in microprocessors still to come, we anticipate future editions of the textbook and video tapes to be issued at regular intervals with new chapters incorporated to cover various technological advances and to maintain a blend of basic principles and current technology. To find out what new material or supplementary texts are available, write to Tom Robbins at the address on page x.

Many people beside the author have made substantial contributions to the textbook and video tapes. The author owes a deep debt of gratitude in particular to John Wakerly for his timely and thoughtful comments throughout the project development. Other manuscript reviewers have added their unique perspectives and have helped to create a better textbook than the author could have done in their absence. Among the many reviewers were Jack Lipovski, Martha Sloan, Jacob Abraham, Ed Bruckert, and Dominique Thiebaut. Through John Fitch's skills in video production and direction, I was able to develop the accompanying video-tape course, but he deserves the bulk of the credit for showing the author the power of the medium and the techniques for tapping its unique capabilities. Tom Robbins, the acquisitions editor for the project, maintained his enthusiasm for the project from our very first phone conversation through the difficult times of final book production. His management activities behind the scenes freed the author from many frivolous problems, and let the author focus his activity on the textbook itself. Marilee Sorotskin's gift for the details of editing and consistency added materially to the quality of the exposition. Finally, the disk operating-system for the word processor on which the book was developed was written by my wife, Jan Stone. Her support both as a spouse and as a live-in systems programmer was extraordinary, and was a critical ingredient in the project development.

Amherst, Massachusetts
May 1982

H. S. S.

READER'S PREFACE

This book is intended for both the undergraduate and the professional reader. Undergraduates should be majors in a computer engineering or enrolled in a computer science program that has exposed them to logic design, assembly-language programming, and a high-level language prior to using this textbook. The professional reader with training in electrical engineering, computer science, or other technical areas is likely to have a sound technical background but needs to brush up on microprocessor technology. This reader will find it useful to browse through the book to learn the major subject areas discussed, and then to concentrate on the unfamiliar material.

Topics are covered three different ways. Each chapter opens with a discussion of basic principles, followed by methods for applying these principles. The chapters close with detailed examples of the principles put to use. The principles are the foundations of microprocessor technology and will continue to be as important in the next decade as they are today. However, the devices available change rapidly as technology advances. The processors, memories, and I/O ports that a student learns about in a sophomore laboratory are obsolete by the time student graduates. Only the principles remain relatively stable within this time frame.

To put this textbook to best use, the reader should learn the principles first, then how to apply them using current technology. The examples in the textbook illustrate practical designs that use real devices available in 1982. Armed with detailed specifications of new devices and with the basic information contained in this textbook, the reader should no difficulty adapting to the most modern devices.

An essential part of the undergraduate learning experience is the experimental lab associated with the textbook. The experiments clarify the principles. Practice in applying the principles can come in the experimental lab or in a later project-design laboratory.

The professional reader undoubtedly has had some laboratory experience. Although having an experimental lab while reading is useful, it may not be necessary for those readers who have older degrees in Computer Engineering or Computer Science and who wish to use this book to bring themselves up to date. These readers should focus on very specific topics. They will probably be able to absorb the material through reading without conducting the lab experiments. Some readers may have sufficient equipment at their disposal in their companies to be able to conduct selected experiments where the experiments are central to the learning process. Several demonstrations are available on color video tapes and may be accessible to the professional through a company library or short-course. In any case, the professional reader should have the experience and maturity to recognize what topics in the textbook must be mastered and to devise a strategy for mastering them.

There are several reader objectives that this book addresses. Readers who are or who wish to become professional designers will find the material to be quite relevant to their work. These readers need a thorough background in electronics and logic design in addition to the material taught in this text. Additional background in transmission lines (for Chapter 2) and linear systems (for Chapter 7) may also be useful. Another group of readers will be concerned with connecting microcomputers to I/O devices or to other microcomputers, and will probably use existing interfaces instead of designing new ones. This group of readers should focus attention on Chapters 1 (microprocessor structures), 5 (serial interfacing), 6 (parallel interfacing), and 9 (software development). Chapter 2 (shielding, grounding, and transmission lines) may be helpful if the reader must specify the physical connections between systems. Chapter 3 (bus interconnections) covers protocols and timing questions that may also be important issues when configuring complex systems.

Another reader of the textbook may be strong in software and relatively weak in electronics. A typical reader of this type may wish to control I/O through software without becoming expert in logic design. This reader will find Chapter 9 (software development) especially illuminating, and will also find topics of interest in Chapters 1 (microprocessor structures), 4 (memories and DMA), 5 (serial interfacing), 6 (parallel interfacing), 7 (magnetic-recording techniques), and 8 (CRT-controller design).

The color video tapes associated with the textbook are an extremely effective way of observing the principles in action. We particularly recommend the tapes for readers who have limited access to experimental equipment because they provide vivid demonstrations of several of the more important experiments. The professional may find a short-course environment with video tapes to be an effective means of learning the material, not only as a way of observing experiments, but as an opportunity to raise and answer questions through class discussions.

I am most interested in the readers' reactions to this textbook. The intended audience is quite broad with diverse skills and backgrounds. Discussions that are over the head of some readers may be too basic and trivial for others. Comments on the strengths and weaknesses of the material in the context of its use are greatly appreciated and may strongly influence future revisions of the material.

Amherst, Massachusetts H. S. S.
May 1982

CONTENTS

1 / MICROCOMPUTER STRUCTURES

This chapter reviews the general characteristics of microcomputer systems. We focus here on the functional description of the major components and on the system structure. By understanding these facets of microcomputers, we will be able to select among several alternative approaches to interfacing and to work easily with both software and hardware to put together complex systems. The functional descriptions in this chapter treat data and control flow exclusively, and ignore specific details of timing and electronics. Later chapters carefully delve into these details and should be sufficient to prepare the reader for practical interface design. This chapter, like those that follow, opens with a general description of the major concepts presented and ends with specific examples to illustrate actual implementations of the concepts.

1.1 BASIC MICROCOMPUTER STRUCTURE

A very simple microcomputer system is composed of three types of modules typically connected as shown in Fig. 1.1. The components are

1. a microprocessor, which contains the control logic and arithmetic unit of the system,
2. a memory, which holds programs and data, and
3. an input/output (I/O) system, which contains one or more ports that connect to such external devices as terminals, disks, printers, and communications modems.

Fig. 1.1(a) shows a single bus system through which the memory and I/O system communicate with the processor. In this system the processor is the master controller. It initiates all activity on the bus by issuing commands to the memory and I/O systems. The bus carries only one transaction at a time, so that commands are issued sequentially, one at a time. The memory and I/O systems respond to the commands, but do not issue commands in turn. If we monitor what happens on the bus over a period of time, we see the processor issuing a sequence of commands, with each command directed to a particular port or memory cell. One type of command instructs the destination to accept data from the processor, and the data from the processor accompanies the command. Another type of command tells the destination to return data to the processor, and the destination replies by sending the requested data back to the processor. So the flow of information is from the processor to memory and I/O, with flow in the reverse direction in response to processor commands.

A slightly different arrangement is shown in Fig. 1.1(b). Here the I/O and memory systems have independent paths to the processor rather than a shared path as in Fig.

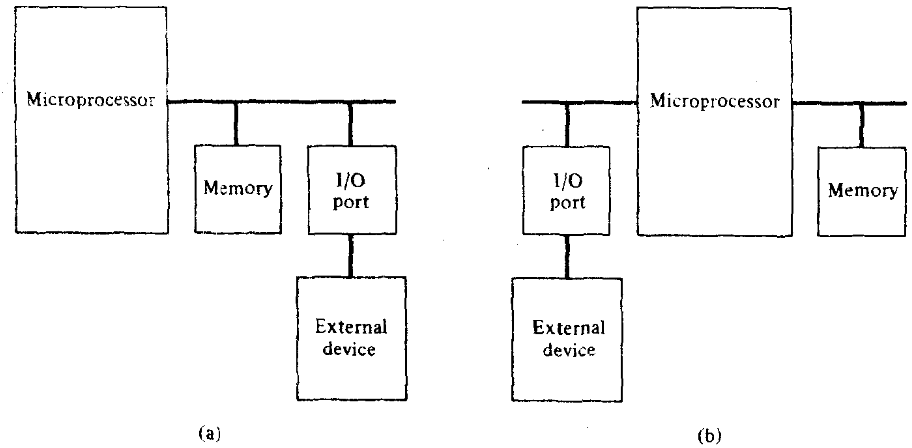

FIGURE 1.1 The basic structure of microcomputer: (a) One-path system; (b) two-path system.

1.1(a). Because the paths are separate and independent, two different transactions can be active at the same time, one on each bus. That is, the processor can issue commands to memory while simultaneously issuing commands to the I/O system. This form of microprocessor has I/O bus control embedded on a chip, together with other conventional microprocessor functions.

The idea of embedding additional functions on chip, such as the ability to control I/O and memory independently, has been carried much further than Fig. 1.1(b) indicates. The I/O ports themselves have been integrated with the processor, so that a microprocessor can be connected directly to external devices and does not need supporting I/O-port chips. In addition, such processors often include a substantial amount of memory integrated with the other functions on one chip. In this form, the microprocessor is truly a single-chip microcomputer, since it contains all of the functions shown in Fig. 1.1(a).

In either of the systems shown in Fig. 1.1, the basic system behavior is the same. The processor interaction with memory is typically a repetition of the sequence below:

1. Fetch an instruction from memory.
2. Execute the instruction, possibly reading data from or writing data to memory.

A special processor register, the *program counter*, controls which instruction to execute next. The execution of each instruction modifies the program counter in a prescribed way, so that when the processor has executed one instruction, the program counter has been updated to indicate a new instruction to execute. The execution of a single instruction, in general, involves one or more additional bus transactions that depend on the instruction executed. For example, the processor may read from or write to memory to exchange data between internal registers and memory. For I/O transactions the processor may obtain status or data from an I/O port or send instructions or data to a port. Even though the processor performs no bus transactions during the execution phase of an instruction, the processor may alter internal registers other than the program counter, which every instruction updates. Figure 1.2 shows an instruction execution graphically, including the relative timing of the memory transactions and the independent activity of the processor.

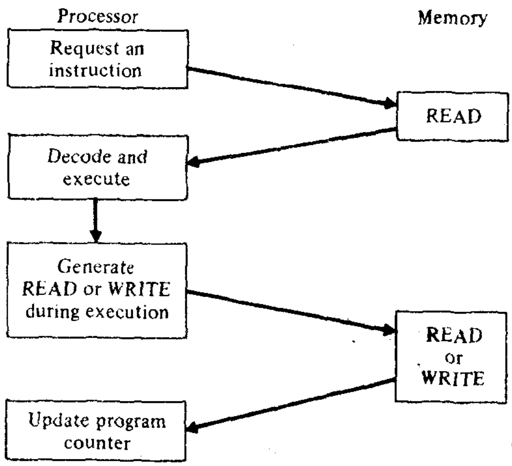

FIGURE 1.2 The time sequencing for processor/memory interactions during the execution of a single instruction.

The I/O interface described thus far appears to behave much like a memory interface. The processor can read data from and write data to memory or to the I/O system using the same type of interface for both memory and I/O. Some computers have a shared common bus for memory and I/O, so for these computer systems the processor-memory interface is essentially the same as the processor-I/O interface. But I/O is somewhat different from memory because of timing and synchronization, in that an I/O interface actually has a superset of the memory-interface functions.

Memory cycles require a fixed maximum time, and memory responds to a processor command within this fixed time. The I/O system, however, has to control or sense events external to the computer whose timing is totally independent of the computer timing. When a processor issues a command to an I/O port to accept a datum from an outside source, that datum can arrive at any time in the future (if it arrives at all), and the state of the processor at the time of arrival is unpredictable. To deal with data moving to and from