INTERFACING TECHNIQUES

In Digital Design

With Emphasis On Microprocessors

Ronald L. Krutz

INTERFACING TECHNIQUES IN DIGITAL DESIGN WITH EMPHASIS ON MICROPROCESSORS

Ronald L. Krutz, Ph.D.

Director, Computer Engineering Center
Mellon Institute
Carnegie Mellon University

WILEY

John Wiley & Sons

New York · Chichester · Brisbane · Toronto · Singapore

Library of Congress Cataloging in Publication Data:

Krutz, Ronald L., 1938–
Interfacing techniques in digital design with
emphasis on microprocessors.

Includes index.
1. Interface circuits. 2. Microprocessors.
3. Digital electronics. I. Title.

TK7868.I58K78 1988 621.398 87-25227

Printed in Singapore

10 9 8 7 6 5 4 3 2 1

INTERFACING TECHNIQUES IN DIGITAL DESIGN WITH EMPHASIS ON MICROPROCESSORS

To Hilda, Sheri, and Lisa

The challenges are simpler—
the failures are fewer—
the rewards are greater

. . . . *because of you*

ABOUT THE AUTHOR

Ronald L. Krutz is Associate Director of Mellon Institute, Carnegie Mellon University. He is also Director of the Mellon Institute Computer Engineering Center. Prior to this position, Dr. Krutz was on the faculty of the Carnegie Mellon Electrical Engineering Department and held senior management and research positions with the Singer Corporate Research & Development Laboratory and Gulf Research Laboratories. He is a Registered Professional Engineer in Pennsylvania and is the author of a videotape course on microprocessors (1984) as well as two other textbooks: *Microprocessors and Logic Design* (1980) and *Microprocessors for Managers* (1983).

PREFACE

Microprocessor interfacing technology is evolving as dynamically as the VLSI components it supports. The techniques of interfacing are based on fundamental electrical engineering principles, advanced microfabrication rules, and computer hardware and software design concepts. It is the aim of this book to tie together these different areas and develop interfacing skills based on practical design experience and meaningful, relevant examples.

Since an increasing number of microcomputer peripheral circuits are being integrated onto higher density chips, a complete coverage of interfacing should include an introduction to basic on-chip device and logic circuit design principles. This approach develops an understanding of the on- and off-chip drive requirements that have to be considered by the designer. Following a discussion of microprocessor interfacing methodologies, terms, and definitions in Chapter 1, the fundamentals of MOS and CMOS logic devices and their interfacing are presented in the first part of Chapter 2. Then transmission line principles and formulas are developed in detail and related to on- and off-chip line-driving applications. Useful transmission line design rules are derived and illustrated with examples using commonly available components. Chapter 2 concludes with a review of noise sources in digital circuits and presents the corresponding shielding and grounding techniques to minimize the associated problems.

Chapter 3 is devoted to the analysis of bus architectures and the interfacing to these buses. Specific sections include detailed coverage of dynamic RAM interface design, EEPROM system design, and HSCMOS/NMOS interfacing. Building on the detail of Chapter 3, the material in Chapter 4 covers serial and parallel interface standards and gives specifications for the most popular of these standards. Detailed examples using available components illustrate the application of serial and parallel interface standards.

As industry standards, the architecture of the IBM PC and the later architectures using the 80386 microprocessor are the main topics of Chapter 5. An introductory summary of the IBM Personal System/2 family is also provided. Through these architectural media, the interfacing material developed in the preceding chapter is illustrated, reviewed, and extended. Details of 80386 memory and I/O interfacing are specifically covered in depth. Included in this coverage are 80386 bus timing, pipelined and non-pipelined cycles, and cache memory interfacing. This approach provides valuable exposure to the student who will more than likely encounter these architectures in many applications.

Realizing that the "real world" is analog in nature, the conversion and interfacing of the analog signals to digital form is explained in detail in Chapter 6. The reverse technology, digital-to-analog conversion, is also developed.

In light of the emphasis on Computer Integrated Manufacturing (CIM) and Computer-Aided Engineering (CAE) in most industries, the topic of local area networks (LAN's) in Chapter 7 is especially relevant. In particular, the ISO-OSI standards and the related MAP standard for manufacturing system communication as presented in Chapter 7 are essential knowledge for the student now and in the future. Also, other token passing and CSMA/CD LAN standards are defined and discussed in depth.

Chapter 8 includes material not usually associated with interfacing—software. In most actual design and interfacing problems, however, software development is the most critical and costly area to be addressed. In particular, software engineering techniques and languages such as C and PL/M-86 that are used to develop interfaces are summarized and relevant examples are presented.

RONALD L. KRUTZ

CONTENTS

1 MICROPROCESSOR INTERFACING: AN OVERVIEW

Interfacing is a term that applies across a broad range of electronic implementations. It relates to systems as well as to individual transistors. Because of this breadth, interfacing involves many technologies that are encompassed by the disciplines of electrical engineering and computer engineering. Since these technologies are usually separated into subdisciplines with their own areas of emphasis, their integration into a reasonably concise and unified treatment is desirable. The treatment of interfacing in this book is directed toward the integration of these various subdisciplines.

1.1 INTERFACING LAYERS

Interfacing usually involves effectively traversing a boundary from one entity to another. In the field of electronics, the entities can be viewed in a hierarchical fashion from a system, subsystem, component, and transistor level. Boundaries have to be traversed in all these levels. In doing so, opposite ends of the spectrum, such as the effects of electrons in motion and instruction execution times, may have to be considered.

In a microprocessor-based system, a number of levels can be defined arbitrarily to attempt to structure the interfacing hierarchy. One possible layering is

electrical(physical)

signal

logic

protocol

code

algorithmic/heuristic

Note that this model includes software implementations as part of the hierarchy. In microprocessor-based systems, especially real-time data acquisition and control applications, the software "interface" to the equipment to be controlled is a costly and critical portion of the total implementation. This aspect, denoted by the code and algorithmic/heuristic levels, is treated in Chapter 8.

Protocol interfacing is studied through communications and local area network standards and implementations. Interfacing at the logic level is addressed through circuit applications, bus interfacing, and data transfer. Signal interfacing techniques are developed in electrical interfacing, bus interfacing, and data transfer techniques. The physical interfacing techniques are developed through the electrical and analog interfacing coverages. All the levels are summarized and illustrated by means of the IBM Personal Computer family and 80386 microprocessor-based material.

1.1.1 OTHER CONSIDERATIONS

In the majority of interfacing applications, timing considerations are critical and must be incorporated into the overall design. In many real-time control systems, the exact response times of all the elements to be controlled may not be available prior to system installation. This unavailability may be due to the sheer number of controlled elements or to their inability to be measured until the total system is assembled and tuned. In such cases, hardware and software accommodations must be made during the specification, design, and implementation phases of the project to handle these timing variables. This "handling" must include the ability to make some changes in the software without requiring radical rewrites or time-consuming modifications of the interface.

Theoretically, since software is more easily changed than hardware, required adaptations to the control system should be made through code changes. If the software was designed using modern software engineering principles, this approach is feasible. Even assembly language coding that is required for time-critical portions of the software can be designed and implemented in a structured manner. Examples of good real-time software implementations for the C and PL/M 86 languages are given in Chapter 8.

Before proceeding to the specifics of interfacing, some fundamental microprocessor-related definitions, characteristics, and economics are provided.

1.2 DEFINITIONS

Strictly speaking, a *microprocessor* is composed of an arithmetic logic unit (ALU), control circuitry, a small amount of scratch pad memory, and, in some cases, timing (clock) circuits. A block diagram of a microprocessor is given in

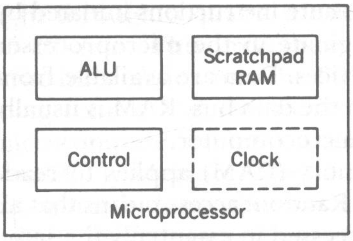

FIGURE 1.1 Microprocessor block diagram.

Figure 1.1. On the other hand, a *microcomputer* is made up of a microprocessor, program memory that is usually a type of read only memory (ROM), read/write random access memory (RAM), and a peripheral interface that provides communication to external devices. If the timing (clock) circuits are not contained in the microprocessor, then they must also be provided for in the microcomputer. A general microcomputer configuration is shown in Figure 1.2.

1.2.1 BUSES

The component elements of the microcomputer are interconnected by wiring paths known as *buses*. There are three types of buses: *address, data,* and *control.* The address bus carries binary-coded patterns (addresses) from the microprocessor to the ROM, RAM, and peripheral interface. The addresses identify unique locations that will be the source or destination of information transferred to or from the microprocessor. This information travels over the bidirectional data bus. The information falls into the four general categories of instructions, control words, status indicators, and data. Instructions contained in ROM are executed by the microprocessor, but intelligent peripheral

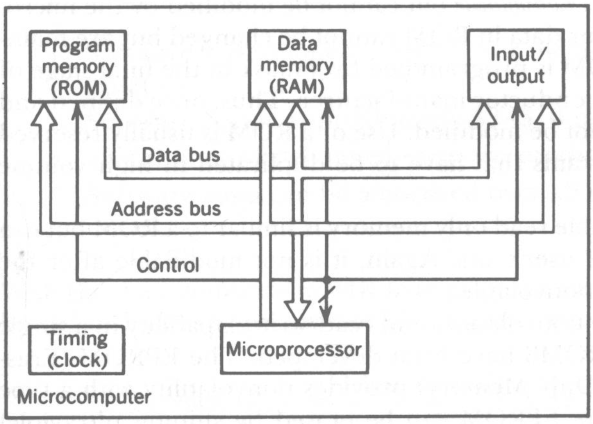

FIGURE 1.2 General microcomputer configuration.