Introduction to Microcomputers and Microprocessors

Introduction to Microcomputers and Microprocessors

ARPAD BARNA

Hewlett-Packard Laboratories
Palo Alto, California

DAN I. PORAT

Stanford Linear Accelerator Center
Stanford University
Stanford, California

A WILEY-INTERSCIENCE PUBLICATION

JOHN WILEY & SONS, New York • London • Sydney • Toronto

Library of Congress Cataloging in Publication Data:

Barna, Arpad.
Introduction to microcomputers and microprocessors.

"A Wiley-Interscience publication."
Bibliography: p.
Includes index.
1. Miniature computers. 2. Microprocessors.

I. Porat, Dan I., joint author. II. Title.
QA76.5.B293 001.6'4'04 75-31675
ISBN 0-471-05051-2

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Preface

The introduction of an ever-increasing variety of microcomputers and microprocessors has led to a great diversity in their applications. The design of microprocessor-based systems, however, involves an understanding of several disciplines, including logic design, digital systems, computer architecture, programming, and to a lesser degree electronic circuit design and semiconductor technology. This introductory book is written for the beginner who does not have a detailed knowledge of these fields but who wants to learn the techniques required for the efficient use of microcomputers and microprocessors.

The subject is presented in three stages. The first three chapters provide an overview of the basic hardware and software, the next five chapters detail the operation, and the last chapter presents additional features. Each chapter can stand on its own, with a minimum of cross-referencing, so that the reader can omit chapters without impairing readability and can conveniently utilize the material for reference.

The 120 examples and problems incorporated in the text make the book particularly suited for self-study, providing a solid basis for understanding the characteristics of the wide variety of available microcomputers and microprocessors. The references listed at the end of the book provide additional material on the subjects discussed. Answers to selected problems are also given.

Arpad Barna
Dan I. Porat

Stanford, California
September 1975

List of Abbreviations

ALU	arithmetic-logic unit
BCD	binary-coded decimal
ASCII	American Standard Code for Information Interchange
CCD	charge-coupled device
CPU	central processor unit
DMA	direct memory access
I/O	input-output
MAR	memory address register
MDR	memory data register
MOS	metal-oxide-silicon
MOSFET	metal-oxide-silicon field-effect transistor
PLA	programmable logic array
pROM	programmable read-only memory
RAM	random-access memory
ROM	read-only memory

Contents

1

Perspective

One of the most significant steps in the development of digital computers was the introduction of the *stored-program computer.** Unlike an abacus or manually operated desk calculator, the operating sequence in a stored-program computer is controlled by an internally stored *program.*

> **Example 1.1.** Vehicular traffic at the intersection of a principal and a secondary road is regulated by a traffic controller that has a 60-second timing cycle. Traffic lights for the principal road are green for 30 seconds, followed by a 5-second amber, a 20-second red, and a 5-second amber. Simple as it is, this traffic controller can be considered to be a stored-program computer.

In current interpretation, however, a stored-program computer has an additional feature: it is capable of *branching* between various segments of its program. Such branching, or *decision-making,* can be controlled by a result of previous computations; it can also be controlled by information received from an *input device* of the computer.

> **Example 1.2.** The traffic controller of Example 1.1 is expanded to include two vehicle sensors that are connected as input devices to the controller. The sensors, located on the secondary road, indicate when a vehicle is waiting for the traffic light to change. At the completion of the 30-second green light for the principal road the controller interrogates the sensors and

* *New terms are italicized.*

changes the lights only if a vehicle is waiting on the secondary road.

Stored-program digital computers have become widespread during the past two decades. This was due primarily to technological developments such as the introduction of transistors that now permeate all parts of the computer, improvements in the storage elements used in the *memory*, increased reliability of the electromechanical *peripheral devices*, and the increasing use of integrated circuits. Present-day digital computers include *special-purpose computers* tailored to a single use and *general-purpose computers* utilized in many diverse areas, such as control, data processing, and scientific calculations.

Parallel with the improved reliability, computational capability, and ease of use of general-purpose computers came the general-purpose *minicomputers* which, though limited in computational capability, were smaller and less costly. Principally because of their lower cost, minicomputers have penetrated into many applications that were previously in the exclusive domain of small special-purpose computers. The remaining gap separating general-purpose computers from special-purpose computers and controllers is being filled by the latest, and smallest, general-purpose computer, the *microcomputer*.

The first microcomputers were calculators. Now microcomputers are also replacing and augmenting many minicomputers and special-purpose computers, particularly special-purpose *hard-wired controllers*.

Example 1.3. Backup safety interlocks are installed in a rapid transit system. A separate interlock is installed for each "block" of the track, monitoring the trains entering and leaving the block. In the initial demonstration each interlock used a special-purpose hard-wired controller. Because of "special cases" arising from various branches in the track, however, the controllers could not all be identical. In the final realization the hard-wired controllers are therefore replaced by microcomputers and the special cases are handled by appropriate programming.

The simplicity and reduced cost that make microcomputers widely applicable also result in programming techniques that are often difficult and clumsy compared to those required by minicomputers. Further, the circuit, or *hardware*, aspects are often more enmeshed with the programming, or *software*, aspects in a microcomputer than in a minicomputer. Thus, although the work may sometimes be divided between "hardware experts" and "software experts," the development of a system that uses a microcomputer frequently requires basic knowledge in both fields. For this reason, the two fields are interwoven in most of this book, aiming to provide a balanced introduction to hardware and software in microcomputer applications.

2

Basic Structure of Microcomputers and Microprocessors

A simplified block diagram of a microcomputer is shown in Figure 2.1. It consists of three functional blocks: the *input-output* (I/O) *section,* the *central processor unit* (CPU), and the *main memory.**

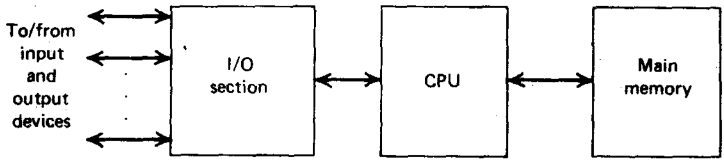

Figure 2.1. Simplified block diagram of a microcomputer.**

2.1. THE INPUT-OUTPUT SECTION

The lines at the left of the I/O section shown in Figure 2.1 connect the microcomputer to the *input* and *output* (I/O) *devices,* also known as *peripheral devices.*

* *A more detailed block diagram appears in Chapter 6.*

** Interconnecting lines in block diagrams may represent multiple connections.

Example 2.1. A handheld calculator has 10 numerical keys labeled 0 through 9, five function keys +, −, ×, ÷, and =, and six digits of decimal display, and incorporates a microcomputer that processes and stores the data. The keys are the input devices and the displayed digits are the output devices.

A simplified block diagram of an I/O section is shown in Figure 2.2. Selection of the I/O devices is performed by *input* and *output* (I/O) *multiplexers* (often abbreviated as MPX or MUX), also known as *data selectors.* Output information is stored in the *output buffers.* The *I/O register* provides temporary storage during the transmission of information between the CPU and the I/O section.

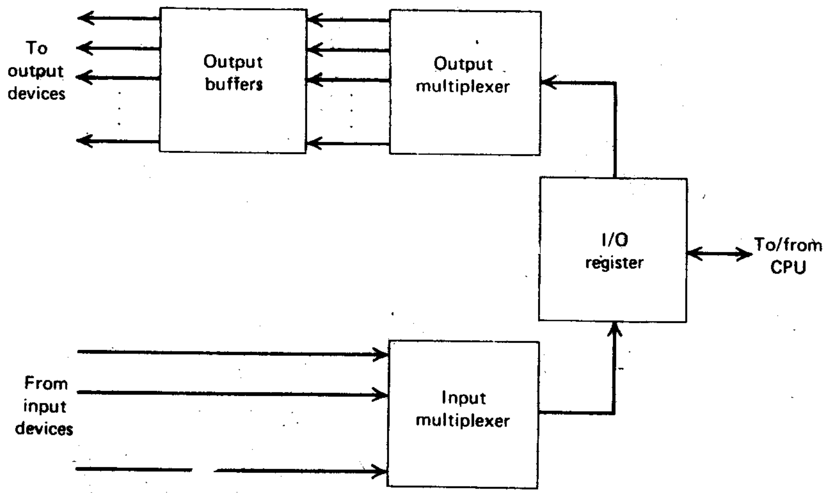

Figure 2.2. Simplified block diagram of an I/O section.

Example 2.2. A traffic controller at a street intersection uses four sensors indicating the presence of vehicles and four traffic lights. The controller incorporates a microcomputer to which the sensors are connected as input devices and the traffic lights as output devices. It is necessary to operate all four traffic lights continuously as green, amber, or red.

Since the speed of the vehicles is limited, a sensor detects a vehicle for at least 0.1 second. Thus the four sensors can be scanned sequentially by the microcomputer as long as each sensor is interrogated at a uniform rate of at least 10 readings/

second. The I/O section of the traffic controller microcomputer can be described as in Figure 2.2.

Characteristics and connections of input and output devices are discussed in Chapter 4.

2.2. THE CENTRAL PROCESSOR UNIT

The internal structure of the CPU varies widely among the various microcomputers. In what follows, we describe a simple CPU. It consists of an *arithmetic-logic unit* (ALU), several *registers*, and a *control unit* as shown in the block diagram of Figure 2.3. The number of lines interconnecting the ALU, the *accumulator* (register A), and registers B and M is determined by the *word length*, which is the maximum number of binary digits ("*bits*") the arithmetic-logic unit can process in parallel.

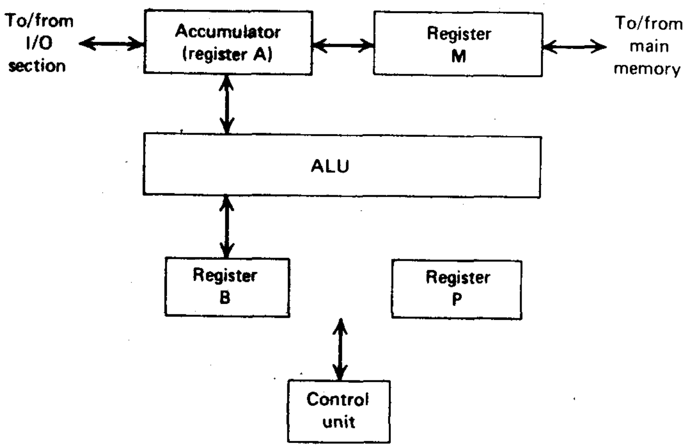

Figure 2.3. Simplified block diagram of a CPU. Connections of the control unit and register P are not shown.

The Arithmetic-Logic Unit

The ALU operates on one or two numbers. It performs arithmetic operations such as addition and subtraction, and logical operations such as the detection of equality. The structure and operation of an ALU are discussed in Chapter 6.

Registers

The CPU includes several registers often designed as *data registers, working registers,* or *scratch-pad memory.* Register A (accumulator) and register B provide storage for data operated on by the ALU.

> **Example 2.3.** In the CPU of Figure 2.3 addition is performed by adding the contents of register B to the contents of the accumulator and placing the result of the addition in the accumulator.

The bit capacity of the accumulator, as well as that of register B, are determined by the word length. In some arithmetic operations the two registers can be used together as a single register of double word length.

> **Example 2.4.** Multiplication in a CPU with a 16-bit word length is performed by multiplying the contents of the accumulator by the contents of register B. The result of the multiplication is a 32-bit number which is stored by placing the most significant 16 bits in the accumulator and the least significant 16 bits in register B.

The CPU of Figure 2.3 communicates with the I/O devices by means of the I/O section and with the main memory through register M.

> **Example 2.5.** A microcomputer incorporating the CPU of Figure 2.3 is used in an industrial temperature controller. The controller measures the temperature by means of five sensors, each of which is interrogated at a uniform rate of 12 readings/minute. The temperature is controlled by an electric heater based on the latest 3 readings of all five temperature sensors.
>
> Data from the sensors are transmitted to the main memory via the I/O section, the accumulator, and register M. Temperature data from the latest 3 readings are operated on by the ALU, requiring further data transfers between the CPU and the main memory through register M. The resulting control information is sent to the heater via the I/O section.

Register P is the *program counter* determining the operating sequence of the microcomputer. The program counter counts up by one unless otherwise commanded. Each step can designate a single operation, such as the addition of two numbers, or a sequence of operations.

Example 2.6. A simple heating system consists of a thermostat, a heater that is turned on or off, and the CPU of Figure 2.3. The desired temperature is loaded into register M from the memory. The control cycle is sequenced by the program counter, that is, by register P. A simplified control sequence is outlined in Table 2.1. It is initiated once every 10 seconds by setting the contents of register P to zero.

Table 2.1. Simplified Control Sequence in the Heating System of Example 2.6

Contents of Register P	Operation
0	Start
1	Transfer the desired temperature from register M to register B
2	Read thermostat and transfer the reading into the accumulator via the I/O section
3	Compare in the ALU the contents of the accumulator with the contents of register B
4	Turn on the heater via the I/O section if the contents of the accumulator are less than the contents of register B
5	Turn off the heater via the I/O section if the contents of the accumulator are greater than or equal to the contents of register B
6	End

The Control Unit

The principal purpose of the control unit in a microcomputer is to provide suitable direction of computer operation.

Example 2.7. Addition of two numbers in the CPU of Figure 2.3 is performed as described in Example 2.3. The control unit first establishes *data paths* for sequentially routing the outputs