OSBORNE/McGraw-Hill

AN INTRODUCTION TO MICROCOMPUTERS

Volume 0
The Beginner's Book

Adam Osborne
David Bunnell

Third Edition

Trademark Acknowledgments

The italicized names are trademarks of the following companies with registered trademarks noted with®:

alphaSyntauri® Syntauri Corporation

ATARI® 400/800® Computer Atari, Inc.

Apple® II, Apple® III, Graphics Tablet® Apple Computer, Inc.

CBM, CBM 2001, CBM 8000 series, PET®, PET® 2001, VIC Commodore Business Machines Inc.

CP/M® Digital Research®

IBM® IBM

Osborne 1 Osborne Computer Corporation

Panasonic® Panasonic

SAM, 820 Xerox Corporation

Softcard, Typing Tutor Microsoft Inc.

The Source (Servicemark) Source Telecomputing Corporation

SuperCalc Sorcim

System X-10® BSR (USA) Ltd.

T/Maker P.Roizen

TRS-80® Model II, TRS-80® Model III, Tandy Corporation

UCSD Pascal Regents of the University of California

WordStar MicroPro International Corporation

VisiCalc® VisiCorp

Z80® Zilog, Inc.

Published by
Osborne/McGraw-Hill
2600 Tenth Street
Berkeley, California 94710
U.S.A.

For information on translations and book distributors outside of the U.S.A., please write OSBORNE/McGraw-Hill at the above address.

AN INTRODUCTION TO MICROCOMPUTERS: VOLUME 0 THE BEGINNER'S BOOK THIRD EDITION

567890 DODO 8976543

ISBN 0-931988-64-0

Cover design by Lorli Willis

AN INTRODUCTION TO MICROCOMPUTERS

Volume 0
The Beginner's Book
Third Edition

AN INTRODUCTION TO MICROCOMPUTERS

Volume 0
The Beginner's Book
Third Edition

Adam Osborne
David Bunnell

Introduction

Microcomputers are destined to be ranked alongside automobiles and television sets as among the most significant inventions of the twentieth century.* By now you've probably seen a microcomputer — perhaps in your local electronics store — and you know about the tiny silicon chips that have made microcomputers possible. What you may not realize is that **within the next few years microcomputers will be as common in our everyday lives as the telephone.**

If you know little or nothing about computers of any kind, this book will get you started, whether you want to buy your own computer, or learn how computers work and how they are built. Accordingly, the book is divided into two sections of three chapters each. Section I (Chapters 1-3) examines the features of microcomputers and related products such as printers and disk drives in enough detail that if you should decide to buy your own system, you can avoid most of the common pitfalls. You will also learn what microcomputers do and how they do it. Section II (Chapters 4-6) is for those readers who want to learn about the inner workings of microcomputers and the

* The text of this book has been printed in **boldface** and lightface types in order to let you bypass information you already know and dwell on information you do not yet understand. **Boldface text summarizes all major subject matter.** When you come across anything in boldface you don't understand, then read the accompanying lightface text for extra information.

technical principles upon which they function. Should you study t second half of the book and find yourself eager for still more inform tion, you may want to move on to the next book in this series, *Introduction to Microcomputers: Volume 1 — Basic Concepts.*

Do you have to understand the internal workings of microcompute in order to use them? Not really. If people had to understand how aut mobiles work in order to drive, the number of automobiles on the ro would be substantially reduced.

However, while automobiles benefit from nearly 100 years engineering refinement, microcomputers are relatively new. Just early automobiles had their quirks, such as hand cranks, toda microcomputers are much less accessible to the average person th models that are still on the drawing board.

Compared with the current crop — the Xerox SAMs, Apples, IB personal computers, and Radio Shack TRS-80s — the first microco puters were really tin lizzies. You couldn't go down to a computer electronics store and buy one; you had to order your microcompu through the mail. Furthermore, you couldn't just plug it in. You had build the computer from a kit, which included such parts as transisto chips, diodes, circuit boards, wires, and, if you were lucky, a reasonal accurate instruction manual. Building a microcomputer required soldering iron, a pair of needle-nosed pliers, and a great deal patience. And once you built your microcomputer, there wasn't mu you could do with it other than program its front panel lights to blink and off in predetermined patterns. To make your microcomputer use required additional interface and memory boards that plugged into t microcomputer's "bus." Once you built these "options," you cou connect your microcomputer to peripherals such as teletype machin computer terminals, and line printers. What began as a $400 kit oft turned into a $10,000 "system" — and even then, you had to do m of the programming yourself, since few programs were then on t market.

The early microcomputer users were commonly known as "co puter hobbyists." Many of them were engineers or programmers w built computers in their spare time. Their fascination with comput was such that they really didn't mind the fact that their microco puters, once built, were virtually useless.

The first generally available microcomputer kit to appear on t scene was the Altair 8800, featured on the January 1975 cover *Popular Electronics* magazine. It was the brainchild of a retired Force engineer, Ed Roberts, who owned a tiny company named MI in Albuquerque, New Mexico.

Some of the early microcomputer owners used their expertise to develop products that propelled the microcomputer from the hobbyist market to the consumer, electronic, and small business markets.

Microdome's "two Steves" — Stephen Wozniak and Steven Jobs — were directly responsible for making microcomputers more accessible when they designed the first Apple computer in Jobs' parents' garage. The Apple I was one of the first microcomputers to combine important components such as memory, intelligence, input, and output on a single circuit board. Its enthusiastic reception at the Homebrew Computer Club, a hobbyist club in the San Francisco Bay Area where Wozniak and Jobs were introduced, convinced the two to upgrade the Apple I to the Apple II and to enter the microcomputer business. The result was Apple Computer Inc., which has become one of the most successful microcomputer companies, with revenues in excess of $100 million per year.

The Apple II and its competitors, the Commodore PET and the Radio Shack TRS-80, were much easier to use than the Altair 8800. These machines came with typewriter-style keyboards and video displays which replaced the rows of lights and toggle switches on the earlier hobbyist models. They also came with the BASIC computer language built in; you could use it the instant you turned on the machine. To use BASIC on an Altair you first had to set toggle switches on the front panel and load BASIC into the computer's memory from a paper tape reader (many early microcomputers were interfaced to surplus teletype machines). The whole process of simply getting to BASIC took 20 minutes to an hour.

The early microcomputer business faced a dilemma not unlike that which confronted the early hi-fi industry. People who purchased the first home hi-fi sets had problems locating an ample supply of records. Music companies were reluctant to invest large amounts of money in records until there were enough potential customers who owned record players. The incentive to buy hi-fi sets was in turn diminished because there weren't enough records on the market.

Microcomputers avoided this initial hurdle much sooner than did hi-fi because they are programmable. In effect, with a microcomputer, you can make your own "music."

Today there are literally **hundreds of companies that develop and distribute programs for microcomputers**. (Programs are generally termed *software.*) Many of these companies were started by computer hobbyists who, once they programmed their microcomputers to do something useful, decided to sell their programs to other users.

Thanks to these programmer pioneers, **you can make extensive use**

of a microcomputer simply by turning it on and loading a pro For example, **with software such as WordStar you can turn microcomputer into a word processor** that lets you edit document ters, books, or manuscripts on the video display. When you finished your editing, you press a few keys and the document pri its final, edited form.

Many of the currently available programs for microcompute in the area of education. Microsoft, a company that produced th microcomputer BASIC, sells a program called Typing Tutor that t microcomputer into an effective typing teacher.

One of the most successful microcomputer software produ VisiCalc, distributed by VisiCorp. Available for the TRS-80, A Atari, and IBM personal computers, **this software turns the scre your microcomputer into a sophisticated analytical tool, an *elect spread sheet*** which has an unlimited number of rows and columr each row or column you can enter numbers, formulas for calcul the results of numbers, or descriptive labels for the data you entere changing other values, your data will be automatically recalcu Thus, **VisiCalc is used by many business professionals as a to developing business and marketing strategies.**

Another way to use your microcomputer is to hook up to a puter network. This requires a device that lets you connect microcomputer to a telephone. By dialing a phone number (a loc in most areas) your microcomputer connects to a central compu large "mainframe" machine. The network asks you for your ac number and your secret code before it lets you proceed further. you are properly connected, you can access any number of service example, The Source network carries United Press Internat newswire stories which you can read on your screen. Most net have games you can play and a message service which lets you inst send and receive electronic mail from any of the other net customers, no matter where they might be located.

A complete discussion of all the programs and network ser available for microcomputers would require a separate book. Still, **are plenty of reasons for many users — perhaps, a majority learn how to program their own microcomputers.** As vast as the rent collection of prepackaged programs is, it touches only a fracti the microcomputer's potential. **One of the reasons microcompute bound to profoundly influence all aspects of society is that their are limited only by the creativity of the people who can pro them.** The ability to program allows you to individualize microcomputer to work exactly as you wish it to work.

To program a microcomputer, you need to learn a *programming language.* In a sense, you teach yourself to communicate with a microcomputer in much the same way that you might teach yourself to communicate with a person from a different country. Many programming languages aren't too difficult because they consist of English words.

The **most popular programming language used on today's microcomputers is BASIC**. This language was created by two Dartmouth professors, John Kemeny and Tom Kurtz, in 1964. Its purpose was to make computers more accessible to students. They accomplished this by making the language "interactive" and by using standard English code words. Interactive means that the computer responds directly and immediately to the input of the user. "Talking" to a computer then becomes similar to talking to a person.

Prior to BASIC, the most common method of programming was done with a process called batch FORTRAN. This was a slow and laborious method. Programs were first punched out on cards which were fed to the computer in one batch. The programmers would often have to come back the next day to see if the program worked correctly. Dartmouth's BASIC was an instant success, and soon it was duplicated on many different computers. Bill Gates and Paul Allen, the founders of Microsoft, implemented BASIC on microcomputers when they created Altair BASIC in 1975. Without their efforts, microcomputers would not have gained in popularity as rapidly as they did. Microsoft BASIC is available on many different microcomputers, and estimates are that more than half a million copies have been sold.

Other programming languages, such as Pascal and FORTH, are popular with microcomputer users. **Many programs are written with assemblers which, although less eloquent and more difficult to learn than so-called "higher language" programs, take full advantage of the microcomputer's capability and require much less memory space.** WordStar and VisiCalc were written with assemblers. A thorough explanation of programming and programming languages can be found in Chapter 3.

There is considerable debate in academic circles about **"computer literacy."** What is meant is that **people need to learn more about how computers work in order to take full advantage of them and to prevent others from abusing them.** But, while **one school of thought advocates "computer literacy," another advocates that computers be taught "people literacy."** This second school is saying that computers should be built to fit in with existing ways that people do things. People shouldn't have to understand so much about computers in order to use them. Computers should instead be made to understand people.

Notice that in this discussion of literacy we have used the wo "computer" instead of "microcomputer." Is there any difference l tween these?

A microcomputer's size is not standard. It can be small enough hold in your hand or as large as a desk top. The computer's logic contained on a single microprocessor chip which we will descr shortly. **A microcomputer functions in the same way as a large co puter.**

Large computers, which may vary from cabinet-sized minico puters to room-sized "mainframe" computers, are more powerful th microcomputers — they **can process larger amounts of numbers data in a shorter time**. Handling all the reservations for a major airl company requires a large computer. Microcomputers can do simi tasks, such as maintaining the schedule of a business executive, l only on a much smaller scale.

Because microcomputers are cheaper than other, larger compute they can be used for many applications that previously weren't co sidered practical for computers. As a consequence, **the impact microcomputers will be to expand the number of jobs that can be co puterized**. Like microcomputers, big "mainframe" computers c store recipe files and play computer games, but they are best used p forming "cost-effective" tasks.

Everyone is aware that computers have already had a big impact society. **During any normal day, your life will be touched by co puters many times.**

For instance, your name is probably on mailing lists maintained computers. A large computer can print thousands of address labels i minute. If typists had to type address labels, you would receive l mail; the senders could simply not afford the cost of the typists.

Credit cards exist because of computers. If accountants did all of t bookkeeping associated with credit card accounts, the cost of accou ing would make credit cards uneconomical.

Consider a previous example — airline ticket reservations. You c walk up to a ticket counter and request a reservation on any flight, an where in the country. The ticket agent is able to tell you instan whether seats are available; if you make a reservation, it is recorded stantly. If another customer, hundreds of miles away, requests the sa seat ten seconds later, the customer will not be sold your ticket. Ov booking may occur, but computers tell the airlines exactly how ma tickets have been sold, and on which flights.

In 1950 there was just one commercially available computer — t ENIAC 1. Originally developed to handle the U.S. Census, t

computer filled up a large room, required special air conditioning, and cost more than $500,000 (that is, 1950 dollars).

Not realizing that less expensive computers could be utilized for a multitude of tasks, some scientists at the time thought a dozen or so "supercomputers" should be built to handle all the data processing needs of the United States. Fortunately the scientists were wrong. In 30 years, the **cost of computing power equivalent to the ENIAC 1 has dropped to less than $10 (1981 dollars)**. In the late 1950s and early 1960s, computers costing $1 million or more started to handle data processing for very large companies, those who could afford the high price. Today an equivalent computer system is available for $2000 to $3000 — cheap enough to do the bookkeeping for the local drugstore.

The most significant development in the first 30 years of computers was the invention of the microprocessor — popularly known as the "computer on a chip." First introduced by Intel Corporation in 1972 as the result of a research project headed by E. M. Huff, the **microprocessor compacts the equivalent of more than 10,000 electronic components (transistors, diodes, and so on) on a chip of silicon about the size of a cornflake. These components, which once filled dozens of circuit boards, are the ones that make up the logic of a computer.**

The technology for jamming an ever increasing number of components into a single package — known as large-scale circuit integration — can be traced further back to the space programs of the 1960s. In order to hurl men out of the earth's atmosphere in sophisticated rocket ships, it was necessary to develop ways to compress electronic components into small, lightweight packages.

But once the United States reached its goal of sending astronauts to the moon and returning them safely, the nation's interest in the further development of space technology declined rapidly. This left thousands of engineers with no jobs and nothing to do except start commercial enterprises based on the new technology they learned while working for the government.

These engineers-turned-entrepreneurs were responsible for creating the huge semiconductor industry centered primarly in Northern California's "Silicon Valley" about 50 miles south of San Francisco. Silicon Valley is the home of Intel, Hewlett-Packard, Apple, and many other high-tech companies.

The microprocessor led the way for the microcomputer. However, while the brains of the computer are being compacted into integrated circuit chips, other components such as memory and interface logic (the part that lets you connect to other devices) are also being

compacted onto chips. When the Altair was introduced in 1975, memory chips could each contain 1000 bytes or units of informatic Today's memory chips contain as many as 65,000 bytes and there is i end in sight. (A *byte* is a unit of memory roughly capable of storing o character, a letter of the alphabet, or one digit of a number.)

At the time of this writing, there were approximately 500,000 wor ing microcomputers. By the time you read this book there will be si nificantly more. New microcomputers are being announced weekl and new microcomputer programs come out daily. Major comput companies such as IBM, Xerox, and Hewlett-Packard have moved in a market that was once the exclusive domain of companies like Apr and Cromemco. Meanwhile, Japanese electronic giants includi Fujitsu, Hitashi, and NEC are poised for major moves into the U. microcomputer marketplace.

Where it will all end is anybody's guess. The Osborne 1 microcor puter, which folds up into a briefcase, may be the forerunner of boc sized microcomputers which we will carry with us almost everywhe we go. Already there are small microcomputers with limited capaci including the Radio Shack Pocket Computer. The April 1981 cover *BYTE* magazine showed a possible future microcomputer wristwatc complete with a full typewriter keyboard and thumbnail-sized flop disks.

What we do know is that there is a microcomputer in nearly ever one's future. The time to start learning about microcomputers is no We hope you find this book to be an informative and entertaini beginning.

About the Authors

Dr. Adam Osborne, President and General Manager of OSBORNE/McGraw-Hill, has worked to produce a growing library of microcomputer books and software. He authored the well-known four volume series, *An Introduction to Microcomputers,* and co-authored titles in the *Programming for Logic Design* series. In 1979 Dr. Osborne explained the microcomputer revolution to the layman in *Running Wild — The Next Industrial Revolution.* He has contributed to many other books, including *Z8000 Assembly Language Programming, PET™/CBM™ Personal Computer Guide,* and *CBASIC™ User Guide.* In 1980 he founded Osborne Computer Corporation, manufacturer of the Osborne 1 personal computer.

He is an international speaker and has contributed numerous articles to journals in the microcomputer industry. Dr. Osborne received his B.S. degree from the University of Birmingham, England, and his Ph.D. in Chemical Engineering from the University of Delaware. He is currently working on his first novel.

Dave Bunnell is a pioneer in the microcomputer industry. He was Vice President of MITS, Inc. when they made the first microcomputer, the Altair 8800. He was the founding publisher of *Personal Computing* magazine, and has written many books and articles on the microcomputer industry. Mr. Bunnell is currently publisher of *PC: The Independent Guide to IBM Personal Computers.*

Other OSBORNE/McGraw-Hill Publications

An Introduction to Microcomputers: Volume 1 — Basic Concepts, 2nd Edition
An Introduction to Microcomputers: Volume 3 — Some Real Support Devices
Osborne 4 & 8-Bit Microprocessor Handbook
Osborne 16-Bit Microprocessor Handbook
8089 I/O Processor Handbook
CRT Controller Handbook
68000 Microprocessor Handbook
8080A/8085 Assembly Language Programming
6800 Assembly Language Programming
Z80 Assembly Language Programming
6502 Assembly Language Programming
Z8000 Assembly Language Programming
6809 Assembly Language Programming
Running Wild — The Next Industrial Revolution
The 8086 Book
PET® and the IEEE 488 Bus (GPIB)
PET™/CBM™ Personal Computer Guide, 2nd Edition
Business System Buyer's Guide
OSBORNE CP/M® User Guide
Apple II® User's Guide
Microprocessors for Measurement and Control
Some Common BASIC Programs
Some Common BASIC Programs — PET™/CBM™ Edition
Some Common BASIC Programs — Atari® Edition
Some Common BASIC Programs — TRS-80™ Level II Edition
Some Common BASIC Programs — Apple II® Edition
Practical BASIC Programs
Practical BASIC Programs — TRS-80™ Level II Edition
Practical BASIC Programs — Apple II® Edition
Practical Pascal Programs
Payroll with Cost Accounting
Accounts Payable and Accounts Receivable
General Ledger
8080 Programming for Logic Design
6800 Programming for Logic Design
Z80 Programming for Logic Design
CBASIC™ User Guide
Science & Engineering Programs — Apple II® Edition
Interfacing to S-100/IEEE 696 Microcomputers
A User Guide to the UNIX™ System
PET™ Fun and Games
Trade Secrets: How to Protect Your Ideas and Assets
Assembly Language Programming for the Apple II®
VisiCalc® Models: Home and Office Companion
Discover FORTH
6502 Assembly Language Subroutines

Contents