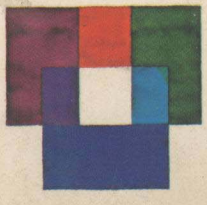
Using the IBM PC: Organization and Assembly Language Programming

Mark Franklin

Assembly language



CBS Computer Books

Using the IBM Personal Computer:

Organization and Assembly Language Programming

Mark A. Franklin

Departments of Electrical Engineering and Computer Science
Washington University
St. Louis, Missouri

HOLT, RINEHART AND WINSTON
New York *Chicago* *San Francisco* *Philadelphia*
Montreal *Toronto* *London* *Sydney* *Tokyo*
Mexico City *Rio de Janeiro* *Madrid*

The cover illustration is the additive color model from Miles Color Art, Tallahassee, Florida prepared using the digital facsimiles process (patent pending), Center for Color Graphics, Florida State University, Tallahassee, Florida.

Address correspondence to:
383 Madison Avenue, New York, NY 10017
First distributed to the trade in 1984 by Holt, Rinehart and Winston general book division.

Library of Congress Cataloging in Publication Data

Franklin, Mark A., 1940–
Using the IBM Personal Computer

Bibliography: p.
Includes index.
1. IBM Personal Computer. 2. IBM Personal Computer—Programming. 3. Assembler language (Computer program language) I. Title. II. Title: The I.B.M. Personal Computer.
QA76.8.I2594F73 1985 001.64 84-10865
ISBN 0-03-062862-8 (pbk.)

ISBN 0-03-062862-8

Printed in the United States of America

Published simultaneously in Canada

4 5 6 039 9 8 7 6 5 4 3 2 1

CBS COLLEGE PUBLISHING
Holt, Rinehart and Winston
The Dryden Press
Saunders College Publishing

Using the IBM Personal Computer:

Organization and Assembly Language Programming

Preface

This book is intended for those who seek to go beneath the exterior view presented by higher-level computer languages, and understand the operation of the IBM Personal Computer (PC) at a deeper level. Accordingly, the book focuses on the architecture and operation of the IBM PC and its machine and macro assembly languages. In addition, we explore the operating system and BIOS (Basic Input/Output System) facilities available at the assembly language level, and we present techniques for interfacing assembly language programs with those written in higher-level languages. Included is all the essential material necessary to program effectively in assembly language.

The Intel 8088 Microprocessor provides the core computing engine for the IBM PC; consequently, the book contains a great deal of material on the operation of this powerful 16-bit microprocessor. A thorough understanding of this material provides a sound basis for understanding the operation of many contemporary computers.

There are numerous practical situations where knowledge of machine organization and assembly language programming is necessary or even essential. These situations generally require that more control be exerted over the internal operations of the com-

puter than is typically afforded by higher-level languages. Such control, for example, may be needed to achieve maximum speed. A higher-level language may be adequate from a logical point of view, but the programs generated may be too slow for the application in question. Close control over the computer at the assembly language level can often exploit the computer's resources in a more efficient manner. Toward this end, special attention is devoted (Chapter 11) to the problem of interfacing assembly language routines with higher-level-language routines. This allows use of the simpler and more powerful programming constructs available in higher-level languages for most programming activity while using assembly language programs for time-critical sections.

Assembly language programming is also needed when close control over standard input/output devices is required, or when special input/output devices are to be interfaced to the computer. Chapter 10 covers the operation and control of several standard devices using BIOS.

Moreover, special applications sometimes require that modifications or additions be made to the computer's overall control program or operating system. Such modifications typically must be done at the assembly language level. Techniques for implementing such adjustments are discussed in Chapter 9, which contains a review of the disk operating system's function calls and their use.

Although the book centers on the IBM PC, the bulk of the material also applies to the IBM PC*jr* and IBM PC/XT computers. All of these systems use the Intel 8088 Microprocessor and therefore employ the same machine and assembly languages. By the same token, the book also can be used to learn assembly language programming on the host of compatible computers now available. This includes systems based on the Intel 8086 Microprocessor, which is faster than the Intel 8088 but functionally identical to it.

This text supports a comprehensive introductory course on computer organization and assembly language programming for people with some background in higher-level-language programming and general notions relating to algorithms and programs. There is no hardware or logic design background required, and any material needed in this area is contained in the text. The book can be used in a formal course setting or as a guide for self-study. Numerous examples are provided in the text, and each chapter concludes with a summary of the main points and a set of problems to help reinforce the material. To take full advantage of the material, an IBM Personal Computer (or compatible equivalent) should be available. The best way to learn assembly language, or for that matter any programming language, is to work problems and write, debug, and run programs.

The book is divided into two major parts: Part I, Chapters 1 through 7, contains basic material on computer organization, operation, instruction set, and assembly language which is needed to write assembly language programs. Part II, Chapters 8 through 11, contains advanced material on macros, interrupts, and DOS (Disk Operating System) and BIOS facilities, with the concluding chapter dealing with higher-level-language–assembly language interfacing techniques. Skill in assembly language programming can be achieved by concentrating initially on Part I, after which more specialized material in Part II can be acquired as needed. Once the language and its associated facilities have been mastered, the book will continue to serve as a useful reference.

I am grateful to my colleagues at Washington University and elsewhere for their support in this endeavor. Particular thanks goes to Sy Pollack for his cheerful encouragement and careful reading of the book, to Tom Patterson for his help in developing and debugging example programs, to Howard Bomze for his close reading of Chapters 9 and 10, and to my brother David for his steadfast optimism that it would get done. My thanks also to CBS Educational Publishing and to the CBS book reviewers who made many valuable suggestions. Finally, I want to thank my wife, children, and friends who had to put up with my replies of the form, "I'm sorry, I have to go work on the book," but nevertheless remained faithful.

The book is dedicated to my wife, Barbara, and to my parents, Jack and Celia Franklin, without whose understanding and help this would not have been possible.

Contents

Part I
The Basics

1

Introduction and Overview

Development of the Large Scale Integration and Very Large Scale Integration (LSI and VLSI) of digital circuits has dramatically lowered the price and increased the performance of computers. This effect, coupled with a large and favorable market response, has resulted in what is called "the personal computer." Computers in this class are *personal* in at least two senses. First, they are inexpensive enough so that many "persons" can now afford to purchase one for themselves. Second, the expectation is that they will often be used in ways determined primarily by the individual user. Their low cost is giving users increased freedom to explore new application areas ranging from use as very smart terminals to dedicated use as processors for a host of other applications.

Often the computer is purchased and used as a packaged, or "turnkey," system where both the hardware and software have been tailored to an application of interest. Dedicated word processing systems fall into this category. In many situations users will program applications for themselves in higher-level languages such as BASIC, FORTRAN, or PASCAL. Such languages significantly ease the task of programming and should be used whenever possible. Use of these languages allows one to concentrate