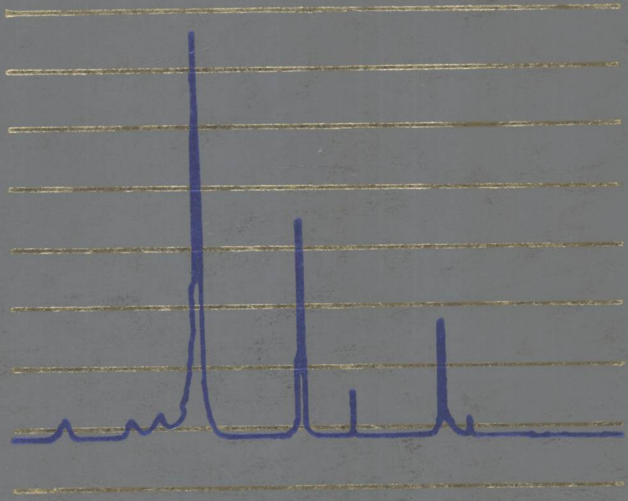
THE LABORATORY MICROCOMPUTER



PROGRAMMING IN PASCAL
AND MC68000 ASSEMBLY LANGUAGE
ON THE IBM SYSTEM 9000

James W. Cooper, Ph.D.

TP31
C17

8563724

TP36
C777-2

THE LABORATORY MICROCOMPUTER

PROGRAMMING IN PASCAL AND MC68000 ASSEMBLY LANGUAGE ON THE IBM SYSTEM 9000

James W. Cooper, Ph.D.

华南工学院
图书馆藏书

E8563724

A Wiley-Interscience Publication
JOHN WILEY & SONS
New York • Chichester • Brisbane • Toronto • Singapore

Copyright © 1984 by John Wiley & Sons, Inc.

All rights reserved. Published simultaneously in Canada.

Reproduction or translation of any part of this work beyond that permitted by Section 107 or 108 of the 1976 United States Copyright Act without the permission of the copyright owner is unlawful. Requests for permission or further information should be addressed to the Permissions Department, John Wiley & Sons, Inc.

Library of Congress Cataloging in Publication Data:

Cooper, James William, 1943–
The laboratory microcomputer.

"A Wiley-Interscience publication."
Includes index.
1. IBM 9000 (Computer)—Programming. 2. PASCAL (Computer program language) 3. Motorola 68000 (Microprocessor) —Programming. 4. Assembler language (Computer program language) I. Title.

QA76.8.I25918C66 1984 001.64'2 84-10437
ISBN 0-471-81036-3

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

THE LABORATORY MICROCOMPUTER

8563724

PREFACE

华南工学院
图书馆藏书

The IBM Instruments CS/9000 provided a new microcomputer system for the scientist, a system that could be programmed in Pascal, a structured high-level language, and that could be used for laboratory data manipulation.

This book gives the scientist a basic introduction to Pascal, shows how to program this computer system, and gives a brief introduction to the assembly language of the Motorola 68000 microprocessor on which this system is based. The book describes algorithms for displays, cursor generation, sorting, and data input and output that the scientist will find convenient to use.

While this book's examples (and indeed its manuscript) were generated on the 9000, much of the book's treatment of Pascal and 68000 assembly language is completely general, and is thus applicable to a number of popular scientific systems now available.

Because the manuscript and all the programs were generated on the CS/9000, all of them have been compiled and run on the 9000 before inclusion in the manuscript, and therefore should run without errors.

This book describes Software Release 1.1 of late 1983. Some reference will be made to earlier versions and to possible future software enhancements.

I first came on the Pascal language while bringing up the UCSD Pascal compiler for Bruker Instruments. At the time I felt that Pascal was a cumbersome "toy" language and not to be taken seriously. However, as I began to work out examples for the instruction manual,

此为试读，需要完整PDF请访问：www.ertongbook.com

I was amazed to discover that they all ran the first time! This single fact probably converted me more directly to a Pascal enthusiast than any discussions of "structure" or logic.

Since this manuscript was written, IBM has changed the computer's name from CS/9000 to System 9000. I have not, however, changed all of the text references, since the computer itself is unchanged.

JAMES W. COOPER

Danbury, Connecticut
August 1984

ACKNOWLEDGMENTS

As usual, a large number of people have made suggestions that have helped to improve the overall quality of this book. Foremost among these are Dr. Doug Dorman and Mary Ann Bogan of Eli Lilly, Inc., and Christine Kostek of IBM Instruments.

I would also like to acknowledge the constant help of Dr. Richard Mushlin, Dr. Willy Shih, Steve Goldband, and Dr. Mel Marple of IBM Instruments for the patient way in which they provided streams of detailed information on a large number of subjects.

Finally, I would like to acknowledge the patience of my wife Vicki and my children Vaughn and Nicole while I wrote and thank PC, our Pussy Cat, for her attentive interest in my work.

J.W.C.

THE LABORATORY MICROCOMPUTER

CONTENTS

PART ONE PROGRAMMING THE SYSTEM 9000 IN PASCAL

1 Introduction to the IBM SYSTEM 9000 3

Files and Filenames 4
Disk and Diskette Volume Names 5
The System Diskette 6
The Keyboard 6
The Command Line 10
Commands and Filenames 10
DIR—List the Directory 11
Specific Files in the Directory 11
Wildcard File Specifications 12
The ? Wildcard 12
Deleting Files 12
The SET Command 13
The RENAME Command 13
Protecting Files: The SECURE Command 13
The SPOOL Command: Printing Files on the Printer 14
SUBMITting Command Files 14
Clearing the Screen 15
Running Programs 15
System Utility Programs 15

COPYing Files 16
FREE Space on Your Diskette 16
Formatting a Diskette 17

2 Editing Programs and Text 18

The Function Keys 19
Starting the Editor 19
Deleting Characters and Text 20
More Ways to Delete Characters 21
Inserting Lines 21
Creating an AUTOEXEC.SUB File 21
Simple Error Correction 22
Storing the File on Disk 23
Other Commands from the Command Line 23
Advanced Cursor Motion Commands 24
Marking and Moving Blocks 25
Searching for Character Strings 26
Locating and Changing Strings 26
Splitting and Joining Lines 28
Aligning Columns with ED 29
An Important Cautionary Note 29
Backing Up Your Files 30

3 Writing a Simple Pascal Program 32

Definition of Pascal 32
Structure in Pascal 33
A Simple Pascal Program 34
The PROGRAM Statement 34
Spacing in Pascal 35
Comments 36
Basic Data Types in Pascal 37
CONST and VAR Declarations 37
Syntax of CONST Declarations 39
Syntax of VAR Declarations 39
The Executable Part of the Program 40
The Assignment Statement 40
Compiling the Program 41
Running the Compiled Program 42
A More Useful Version of ADD2 43
Errors in Compilation 44

Restrictions on REAL Numbers 44
Summary 44

4 Pascal Data Types and Techniques 46

Ranges of INTEGER, LONGINT 46
Hexadecimal Numbers 47
REAL Numbers 47
DOUBLE Precision Numbers 47
Arithmetic Operations in Pascal 48
Conversion Between REAL and INTEGER Types 49
Using INTEGERs and LONGINTs In Expressions 49
Other Common Arithmetic Operations 50
Arithmetic Priority 51
The Address of @-Operator 51
Boolean Types and Operations 52
Variable Names 52
Case Conventions 53
The CHAR and STRING Types 54
Apostrophes Within Strings 54
Internal Representations of Characters 55
Incrementing and Decrementing Character Variables 55
Linking Programs Efficiently with ALINK 56
Creating a SUBMIT File for Simple Compilations 57
Using the "&" Character in SUBMIT Files 57

5 Decision Making in Pascal 59

The IF-THEN Statement 59
The IF-THEN-ELSE Statement 60
The Semicolon in the IF-THEN-ELSE Statement 61
Block Structures in Pascal 61
Program Layout 61
Multiple IF-THEN Statements 62
The CASE Statement 62
Counting Characters 63
The OTHERWISE Clause 64
Blocks in CASE Statements 65
Arrays in Pascal 65
Strings as Arrays 66
Putting Characters into Strings 67

此为试读，需要完整PDF请访问：www.ertongbook.com

Copying Array Values 67
Arrays of Strings 68
Looping—The FOR Statement 68
Using FOR with CASE to Count Characters 70
The REPEAT Statement 71
Comparing REAL Numbers 73
The WHILE-DO Statement 74
Initializing the WHILE Conditions 75
Comparison of the Looping Statements 76

6 Input, Output, and File Handling in Pascal 77

Reading from the Terminal—READ and READLN 77
Formatting WRITE and WRITELN Statements 79
Opening a TEXT File to READ 80
Reading Data from TEXT Files 81
Opening New Files and WRITEing to Them 81
File Variable Names 81
The EOLN and EOF Functions 82
Closing Files in Pascal 83
The File Window Variable in Pascal 84
Interactive Files 85
Reading Single Characters from the Keyboard 86
The CS/9000 Devices 86
Printing Files on the Printer 87
Reading Other Types of Files 88
Untyped Files 88
IORESULT—Input and Output Errors 90
Closing Files and Devices 91
Compiler Comments 91

7 Defining Types in Pascal 93

The Subrange Type 93
Enumerated Types 94
Sets in Pascal 96
Set Operations 96
Records 97
Individual RECORD Elements 98
Shorthand for the Same Record—
The WITH Statement 99

Files of RECORDs 100
Records with Variant Parts 100
Using the Variant Record to Change Types 102
PACKED Arrays 103

8 Procedures and Functions 104

Arguments to Procedures 106
Type Restrictions on Procedure Arguments 106
Value and Reference Parameters 107
Local and Global Variables 108
Scope of Variables 108
Internal FUNCTIONs in Pascal 110
Writing Your Own Functions 111
Calling Your Own Functions 112
Recursion 112
Side Effects 114
A Forgiving REAL Input Routine 114
FORWARD Declarations 117

9 The Pointer Type 119

DISPOSE—Reallocating Memory 120
MARK and RELEASE 121
The Linked List 121
Linking Successive Records Together 122
Scanning Through the Linked List 123
Inserting Records in a Linked List 126
The @-Operator and Pointers 128
Using the @-Operator to Subvert Array
Size Checking 128

10 The GOTO and EXIT Statements 130

The GOTO Statement 130
Syntax of the GOTO Statement 130
Test Data for Matrix Inversion 133
Using the GOTO Statement to Detect Singularity 134
Statement Order 135
The EXIT Statement 136

11 Calling CS/9000 Library Functions 137

Clearing the Screen—CLS 137
Getting Filenames from the Command Line 138
Reading the Keyboard Function Keys 139
Using the AND Function to Check Flag Bits 141
Placing the Cursor with the GOTOXY Function 142
Reading the Date and Time 144
Sending Function Packets to Devices 144
The Keyboard Functions 145
Detecting Ctrl/Break 147
The Printer 149
The SREAD and SWRITE Commands 152
Using the Printer in Graphics Mode 153
The Keypad and Softkeys 154
The Beeper and LED's 155
Adding a Command to the Table 158
The RS-232 Channels 157
Closing Devices Opened in Programs 159
Common Error Codes Returned by SYSFUNC 160

12 The Display Screen: Displaying Characters and Graphics 161

The Display Devices 161
Windows 163
Screen and Console Box Functions 163
Displaying Two Windows on the Same Screen 164
Using the Console Box 166
The Graphics Display Functions 169
Drawing Spectra Using the Graphics Functions 170
Displaying Large Characters on the Screen 174

13 Pascal UNITs: Compiling Routines Separately 176

The Pascal UNIT 176
Error Handling 176
The GRAFUNC UNIT 177
Linking Your Program With UNITs 179

PART TWO PROGRAMMING THE MC68000 PROCESSOR

14 Introduction to the MC68000 183

Data and Instructions 183
Bytes, Words, and Long Words 184
Binary and Hexadecimal Numbers 185
Addition and Subtraction 186
Place Values in a Hexadecimal Number 187
Hexadecimal Arithmetic 188
Getting Used to Hexadecimal Notation 188
Logical Operations in a Binary Computer 189
Complements 189
Two's Complements 190
Subtraction 191
Registers in the 68000 192
The Stack Pointer 193
The Program Counter 193
The Status Register 194
User Byte 194
Supervisor Byte 195
References 195

15 Addressing Modes in the MC68000 196

The MOVE Instruction 196
Comments 198
Format of the MOVE Instruction 198
The Addressing Modes 199
Classes of Addressing Modes 199
Register Direct Addressing Modes 201
Indirect Addressing Modes 201
Auto Increment Indirect Addressing 202
Predecrement Indirect Addressing 202
Indirect Addressing with Displacement 203
Indirect Indexed Addressing 204
Short and Long Absolute Addressing Modes 206
Labels in Assembly Language 207
The PC-Relative Mode 208

PC-Relative Mode with Indexing 208
Immediate Mode 209
Getting the Address of a Location 210
Summary 210

16 Arithmetic Operations in the 68000 211

The MOVE Instruction Group 211
The MOVEA Instruction 212
Special Data Movement Instructions 212
Moving To Special Registers 212
The User Stack Pointer 213
Saving Registers with the MOVEM Instruction 214
The Addition Instructions 215
The Special Addition Instructions 215
Multiple Precision Addition 216
The Subtract Instructions 217
The Logical Operations 217
Single Operand Arithmetic Instructions 218
The Multiply and Divide Instructions 218
Shift Instructions 219
Logical Shifts 221
Rotates 221
Rotates with Extend 221
BCD Instructions 221

17 Decision Making Instructions: Writing Simple Programs 222

The CMP Instructions 222
The TST Instruction 223
Examining the Condition Codes 223
A Simple Program 225
Adding Three Numbers Together 225
Assembler Directives 226
A Program to Add 10 Numbers Together 227
Sign Extension 229
Using the CMP Instruction 230
Decrement and Branch Instructions 231
Using Both Tests of the DBcc Instructions 232
Testing and Changing Individual Bits 233
Assembling Simple Test Programs 234
Using the Assembler 234

Errors in Assembly 235
Linking the Assembly Language File 235

18 Using the Stack for Subroutines and Addresses 236

Subroutines 237
A Simple Subroutine Example 238
Calling Assembly Language Procedures from Pascal 240
Nested Procedures 243
Returning Values from Functions 243
Assembling the Module 243
The LINK and UNLK Instructions 244

19 Advanced Assembly Language Techniques 246

Using the Displacement Indirect Mode 247
PC-Relative Addressing in Source Modes 248
Storage of All Results on the Stack 249
Debugging 252
Examining and Changing Memory Locations 253
Offsets in Modules 253
Breakpoints 255
Exiting from DEBUG 256
Real Number Format 257
Reference 259

20 Using the ASM Assembler 260

Labels 260
Comments 261
Numbers and Constants 261
Symbol Expressions 262
Assembler Directives 263
Storage Definition Directives 264
Using the OFFSET Directive 264
Linker Directives 265
Listing Control Directives 265
Setting Assembler Options 266
Using the Assembler 267
Conditional Assembly 268
MACROs 269
Structured Control Statements in ASM 270

此为试读，需要完整PDF请访问：www.ertongbook.com