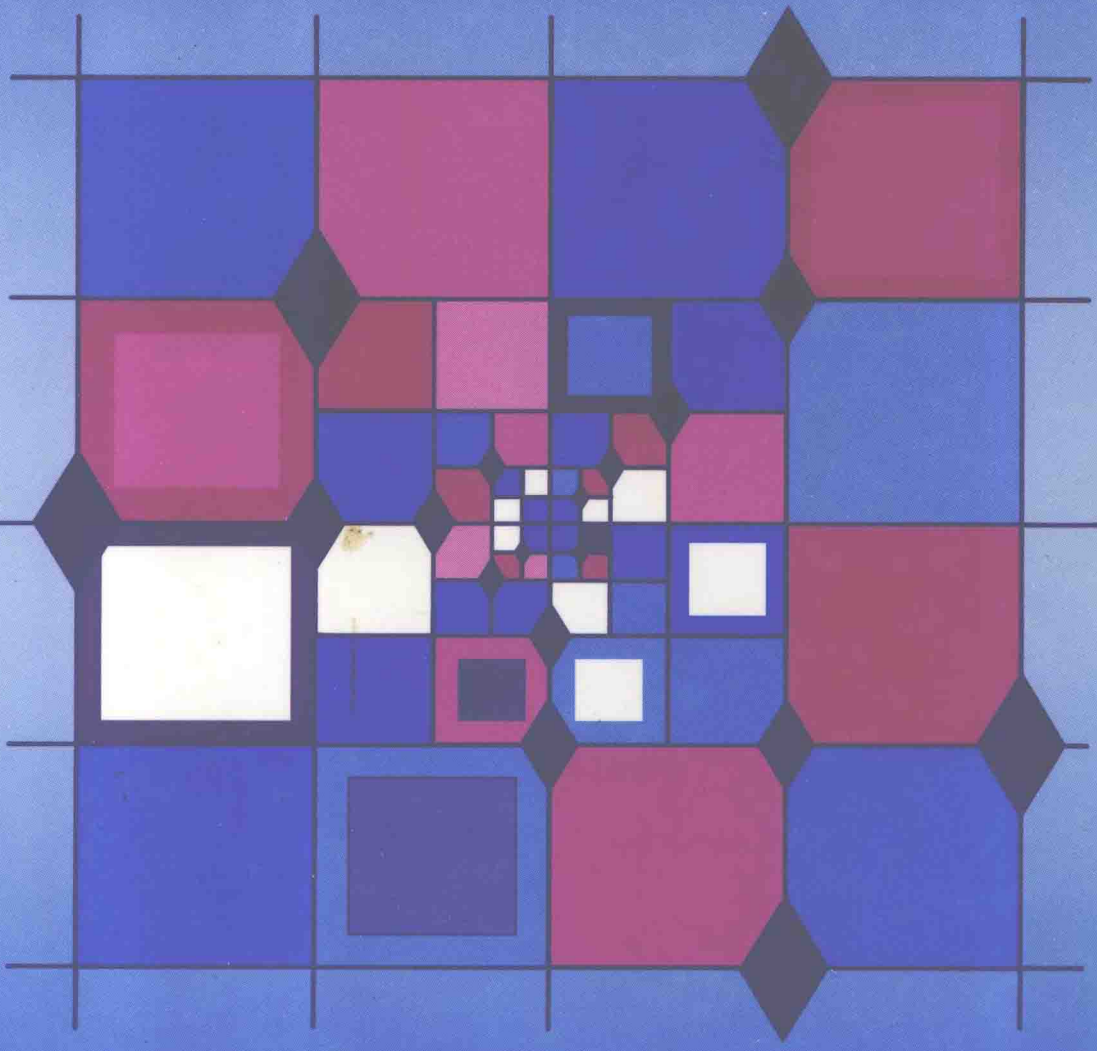
Structured Assembler Language for IBM Computers



Alton R. Kindred

Structured Assembler Language for IBM Computers

Alton R. Kindred
Manatee Community College

Harcourt Brace Jovanovich, Publishers
and its subsidiary, Academic Press

San Diego New York Chicago Austin Washington, D.C.
London Sydney Tokyo Toronto

ISBN: 0-15-584070-3
Library of Congress Catalog Card Number: 86-70503

Printed in the United States of America

Preface

As a student of computer science, you probably learned programming through a high-level language, such as Pascal, BASIC, FORTRAN, or COBOL. You also realize that much of the application programming in the business world is done in high-level languages. You might then wonder, "Why should I study assembler language?"

Benefits of Assembler Language

Assembler language provides a broad understanding of machine functions and processing techniques that cannot be matched through any other source. It offers alternative choices of instructions for data manipulation and calculations. It helps you to understand why certain file and data definitions cause more efficient operations than others do. You may find that subprograms written in assembler language can be called from high-level languages to perform specific tasks not otherwise possible.

Through assembler language you can gain a detailed understanding of address modification, register operations, table handling, and other functions common to many languages. Moreover, you can learn to analyze object code and diagnose storage dumps, both valuable tools in debugging.

Some people contend that assembler language should be avoided because programs must be rewritten every time the hardware is changed. But the IBM family of computers starting with the System/360 and progressing through the 370, 3030, 3080, and 4300 series—as well as a large group of IBM-compatible computers—has used the same assembler language for over 20 years. Although the language has been expanded, programs written in the 1960s can usually be run with few or no changes on hardware that has changed dramatically over the years. Few high-level languages have been so stable.

Shortcomings of Other Textbooks

Many assembler textbooks have certain shortcomings that I have tried to overcome here. They present too much detail early in the text, before the student has an opportunity to assimilate and apply it. They wait too long to provide enough information to produce a complete program. They show too few complete programs—and these usually appear late in the book. They depend

on artificial tools, such as special housekeeping and input/output macros, and often ignore the use of disk files entirely.

But the most serious omission in many assembler books is lack of attention to the principles of structured programming. Students who have learned proper structure in their introductory courses are perplexed at the disregard of proper branching and modular design they often find in their assembler texts.

Features of This Book

This book employs many techniques to design and write correct programs. Structured programming is demonstrated and practiced throughout. Initially, each program is shown in flowchart form, pseudocode, and assembler language. Later the flowcharts are eliminated.

A new method is introduced by which each type of program structure—sequence, repetition, and selection—is clearly identified by the use of names such as IF, THEN, ELSE, ENDIF, DOWHILE, ENDDO, REPEAT, and UNTIL. Consistent use of these names helps to ensure proper structure.

Page heading subroutines are introduced with the first program and other subroutines are emphasized throughout as ways to improve and extend modular structure.

Programs begin with simple character operations and gradually move on to decimal arithmetic, editing, binary arithmetic, address modification, and table handling. Later chapters, which may be omitted without loss of continuity, deal with subprograms, advanced register operations, macro definitions, and floating-point numbers.

No special non-system macros are used. File definitions and processing macros for both DOS and OS are provided as a model that can be used for each of the programs.

Data types and machine instructions are introduced only as needed and are immediately utilized in examples and sample programs. Detailed summaries are presented in appendixes.

Early sample programs are based on the same data set to give continuity and relevance.

File creation and processing are covered with sequential, indexed sequential, and VSAM files. Sample programs illustrate each form of processing for each type of file.

Organization of the Text

The book contains 18 chapters and five appendixes. The intent is always to work from familiar concepts and material to new ones. The first seven chapters should be taken in sequence. Chapter 1 reviews basic hardware and software concepts, with special emphasis on structured program development. Chapter 2 covers types of assembler language statements, file and data definitions in both DOS and OS, and procedural instructions to permit develop-

ment of a complete program utilizing a heading subroutine. Chapter 3 develops more complex program logic while continuing to work with data in character form.

Chapters 4 and 5 introduce packed decimal numbers, arithmetic operations, and editing. Standards modules are developed for reading records and printing lines.

Chapter 6 introduces more technical material on base-displacement addressing, machine instruction formats, the condition code and branching, and input/output channels.

Chapter 7 covers techniques to avoid errors in programs and detect those that do occur. Output from the assembler and linkage editor is examined, and detailed analysis of a core dump is explained.

Chapter 8 presents control break processing, with single-level and multiple-level totals. This chapter may by studied in sequence or deferred until later, when magnetic tape and disk files are covered.

Chapters 9 and 10 deal with operations on binary numbers, address modification, and table handling. Conversion between decimal and binary numbers, arithmetic operations, shifting within registers, and creating and searching tables are treated in detail.

Chapter 11 shows subroutines and subprograms as means of expanding modular structure of programs. Chapter 12 explains macro definitions and conditional assemblies as ways to simplify programming for definitions and routines that are frequently used.

Chapter 13 presents advanced logical operations for working on individual bits within a byte, translating data from one coding structure to another, or scanning strings of characters. This chapter can be studied immediately after Chapter 10 if desired.

Chapters 14 and 15 discuss sequential files on magnetic tape and disk. Topics include fixed-length and variable-length records, blocked records, volume and file labels, disk track capacity, and processing techniques.

Chapters 16 and 17 cover indexed sequential and VSAM files. Either or both may be omitted without affecting any of the book's other contents.

Chapter 18 concludes the text with a discussion of floating-point data, arithmetic operations, and conversion from fixed-point to floating-point data.

Each chapter contains a statement of objectives, numerous illustrations, and (except for Chapter 1) one or more complete illustrative programs. Each chapter concludes with a summary, a list of terms for review, questions for discussion, exercises, and programming problems.

The appendixes include a glossary, a summary of machine instructions, the IBM System/370 reference summary, data for programming problems, and answers to exercises.

Supplemental Materials

An instructor's manual is available with teaching suggestions and a large bank of test questions.

The following IBM publications will provide valuable additional information:

A22-6821 System/360 Principles of Operation
GA22-7000 System/370 Principles of Operation
GA22-7070 IBM 4300 Processors Principles of Operation
GC24-5099 OS/VS1 JCL Reference
GC24-5103 OS/VS1 Supervisor Services and Macro Instructions
GC33-4010 OS/VS-DOS-VS-VM-370 Assembler Language
GC33-4021 OS/VS-VM/370 Assembler Programmer's Guide
GC33-4024 Guide to the DOS/VSE Assembler
GC33-5372 DOS/VS Data Management Guide
GC33-5373 DOS/VS Supervisor and I/O Macros
G320-5774 VSAM Primer and Reference

Acknowledgments

Many people have contributed to the development and production of this book. My students at Manatee Community College have field-tested many of the structured concepts evolving over the past three years. My colleagues in the computer science department and the computer center have offered valuable support and assistance.

The manuscript was reviewed by Joseph S. McFarland, Dundalk Community College, who offered numerous helpful criticisms and suggestions. Dale Brown, of Academic Press, and Ralph Zickgraf, of EDP, Inc., lent their expertise in technical production.

Special thanks, as always, are reserved for my loving wife, Joy, who gave her unfailing inspiration and support throughout the long hours of creation and development of the book.

Alton R. Kindred

Contents

1. Basic Computer Concepts 1

THE COMPUTER SYSTEM 1
HARDWARE 2
- Main Storage 2
- Central Processing Unit 3
  - Control Unit 3
  - Arithmetic-Logic Unit 4
- Input/Output Devices 4

SOFTWARE 5
- The Operating System 5
  - DOS 7
  - OS 9
- Application Programs 10

STRUCTURED PROGRAM DEVELOPMENT 10
- Defining the Problem 10
- Planning the Logical Solution 12
  - Sequence Structure 13
  - Repetition Structure 14
  - Selection Structure 16
- Coding and Assembling the Program 16
  - The Assembly Process 18
  - Relocating the Program 19
- Testing the Program 19
- Completing the Documentation 20

DATA CODES 21
- EBCDIC Code 21
- Hexadecimal Numbers and Addresses 22

2. Getting Started in Assembler Language 30

THE CODING FORM 30
TYPES OF STATEMENTS 32
- Machine Instructions 32

Assembler Instructions 32
Macro Instructions 34
PARTS OF A PROGRAM 34
File Definitions 34
DOS 35
OS 37
Data Definitions 38
Defining Constants 38
Defining Storage 39
Procedural Instructions 41
PROGRAM 1: READING AND PRINTING RECORDS 44
Housekeeping and Initialization 45
The Heading Subroutine 49
The Priming Read 49
The Mainline Loop 49
The End-of-File Routine 51
OS Version of Program 1 51

3. More Operations with Characters 59

EXPLICIT LENGTHS IN MOVING CHARACTERS 59
RELATIVE ADDRESSING 60
MOVING OVERLAPPING FIELDS 60
LITERALS 62
IMMEDIATE INSTRUCTIONS 63
DEFINING HEXADECIMAL CHARACTERS 64
COMPARING CHARACTERS 65
CONDITIONAL BRANCHING 65
NESTED IF STATEMENTS 66
COMPOUND CONDITIONS 67
PROGRAM 2: PRINTING SELECTED RECORDS 69
PROGRAM 3: PRINTING SELECTED RECORDS 70

4. Packed Decimal Addition, Subtraction, and Editing 79

ZONED DECIMAL DATA 79
PACKED DECIMAL DATA 80
The Pack and UNPK Instructions 83
Defining Packed Decimal Data 84
DECIMAL ADDITION, SUBTRACTION, AND COMPARING 85
EDITING DATA 88
The Edit Pattern 89
Zero Suppression 89

Inserting Characters 90
AN ALTERNATE EDITING APPROACH 91
The Unpack (UNPK) Instruction 91
The Move Zones Instruction 92
PROGRAM 4: ACCUMULATING TOTALS 93

5. More Decimal Operations and Modular Structure 100

MULTIPLICATION 100
Aligning Decimal Points 101
The Multiply Packed (MP) Instruction 102
Rounding 103
Dropping Excess Digits 104
Move Numeric (MVN) Instruction 104
Move with Offset (MVO) Instruction 105
Rounding with the SRP Instruction 107
DIVISION 108
THE ASA CONTROL CHARACTER 110
Printing Heading Lines 111
Printing Detail Lines 112
Printing Total Lines 113
STANDARD MODULE TO READ RECORDS 113
PROGRAM 5: ACCUMULATING TOTALS AND AVERAGING 115

6. Computer Architecture 125

GENERAL REGISTERS 125
BASE-DISPLACEMENT ADDRESSING 126
Program Status Word 127
Explicit Register Addressing 128
Multiple Base Registers 129
INSTRUCTION FORMATS 131
RR Instructions 131
RX Instructions 133
RS Instructions 134
SI Instructions 134
SS Instructions 135
S Instructions 135
Instruction Length Codes 135
CONDITION CODE AND BRANCHING 136
INPUT/OUTPUT CHANNELS 138
PROGRAM 6: ANNUAL TOTALS BY PAY TYPE 140

7. Diagnostics and Debugging 149

AVOIDING ERRORS IN PROGRAMS 149
DETECTING ERRORS AT ASSEMBLY TIME 151
- Source Statement Listing 152
  - The Location Counter 155
  - Object Code 155
  - ADDR1 and ADDR2 156
  - The Literal Pool 156
  - Diagnostics and Statistics 156

OTHER ASSEMBLER OUTPUT 159
- External Symbol Dictionary 159
- Relocation Dictionary 159
- Cross-Reference Listing 161

PROGRAM RELOCATION AND LINKAGE 161
- The Linkage Editor Map 165
- Input/Output Modules 165

DETECTING ERRORS AT RUN TIME 166
- System Diagnostic Aids 167
- Analyzing the Core Dump 170

8. Control Break Processing 177

CHECKING SEQUENCE 177
CONTROL BREAK PROCESSING 178
- Single-Level Totals 179
- Multiple-Level Totals 180

PROGRAM 8: TOTAL MEN AND WOMEN BY JOB CODE WITHIN DEPARTMENT 180
CONTROL BREAK PROCESSING WITH IF-THEN-ELSE STRUCTURE 187

9. Binary Operations 192

BINARY NUMBER REPRESENTATION 192
- Unsigned Binary Numbers 193
- Signed Binary Numbers 193

DEFINING BINARY DATA 195
- Binary Constants 195
- Fullwords 195
- Halfwords 195
- Address Constants 196

MANIPULATING BINARY DATA 196
- Loading and Storing Fullwords or Halfwords 197

Loading Addresses 198
Loading and Storing Multiple Registers 198
Other Load Instructions 198
CONVERSION TO AND FROM BINARY 200
BINARY ADDITION AND SUBTRACTION 201
Overflow in Binary Arithmetic 201
Fullword Addition and Subtraction 204
Halfword Addition and Subtraction 204
Addition with Load Address Instruction 205
Subtraction with Branch on Count Instruction 206
BINARY COMPARISON 206
SHIFTING BINARY NUMBERS 206
Single Register Shifts 207
Double Register Shifts 208
BINARY MULTIPLICATION 209
Fullword Multiplication 209
Halfword Multiplication 211
BINARY DIVISION 211
Rounding Binary Numbers 213
PROGRAM 9: AVERAGE SALARY BY DEPARTMENT AND ORGANIZATION 214

10. Address Modification and Table Handling **222**

INITIALIZING ADDRESSES 223
MODIFYING ADDRESSES 223
EXPLICIT OPERANDS 224
INDEX REGISTERS 225
Branch on Index High Instruction 227
Branch on Index Low or Equal Instruction 227
DUMMY SECTIONS 228
DEFINITION OF A TABLE 230
TABLE ELEMENTS 231
CREATING THE TABLE 232
Defining Constants 232
Reading Table Records 232
The ORG Statement 233
SEARCHING TECHNIQUES 233
Factor Matching 235
Address Calculation 235
SINGLE-LEVEL TABLES 236
MULTIPLE-LEVEL TABLES 237
PROGRAM 10: LOCATING INSURANCE RATES 237

11. System Macros, Subroutines, and Subprograms 248

SYSTEM MACROS 248
- COMRG Macro 248
- GETIME Macro 249
- TIME Macro 250

SUBROUTINES 251
- Linkage 251
- Calling Sequence 251
  - Address in Register 252
  - Parameter List 252

SUBPROGRAMS 254
- EXTRN and ENTRY Statements 255
- Register Conventions 255
- The CALL Macro 256
- The SAVE and RETURN Macros 257
- Base Registers for Subprograms 259
- Subprogram Examples 259
- Multiple-Level CALLS 261
- Using Files in Subprograms 262
- Assembler Subprograms with Other Languages 263

PROGRAM 11: TIMING OPERATIONS 263

12. Macro Definitions and Conditional Assemblies 272

INVOKING MACROS 273
DEFINING MACROS 279
FORM OF THE MACRO DEFINITION 274
- The MACRO Header 274
- The Prototype Statement 274
- The Body 275
- The MEND Statement 275

TEXT INSERTION 275
TEXT MODIFICATION 276
- Variable Symbols 276
  - Positional Parameters 277
  - Keyword Parameters 278
- Concatenation 278
- Attributes 278
  - Length 279
  - Type 279
  - Count 281
  - Number 281

Integer 281
Scaling 281
Sublists 282
TEXT MANIPULATION 282
Conditional Assembly Language 283
SET Symbols 283
SETA Symbols 283
SETB Symbols 284
SETC Symbols 284
Sequence Symbols 284
Branching Statements 285
AIF 285
AGO 286
ACTR 286
ANOP 286
System Variable Symbols 286
&SYSLIST 286
&SYSNDX 287
&SYSECT 287
&SYSPARM 288
&SYSTIME 288
&SYSDATE 288
SEVERAL USEFUL MACROS 288
PROGRAM 12: SAV13 AND RET 13 MACROS 291

13. Advanced Logical Operations **299**

BIT MANIPULATION 299
OR Instructions 300
AND Instructions 300
EXCLUSIVE OR Instructions 301
Test Under Mask Instruction 302
ADVANCED EDITING 304
Multiple Field Editing 304
Check Protection 304
Editing Negative Amounts 305
Edit and Mark Instruction 305
LOGICAL SHIFT INSTRUCTIONS 307
Single Register Shifts 307
Double Register Shifts 307
OTHER LOGICAL OPERATIONS 308
Data Movement 308
Addition and Subtraction 310
Comparison 311
Supervisor Call 312

TRANSLATE INSTRUCTION 313
TRANSLATE AND TEST INSTRUCTION 314
EXECUTE INSTRUCTION 315
PROGRAM 13: SCANNING CHARACTER STRINGS 318

14. Magnetic Tape Files **325**

TAPE CHARACTERISTICS 325
- Parity Bit 326
- Longitudinal Check Byte 326
- Interrecord Gap 326
- Tap Density 326
- Tape Speed 326

BLOCKED RECORDS 327
VARIABLE-LENGTH RECORDS 327
TAPE LABELS 328
MAGNETIC TAPE FILE DEFINITIONS 332
- DOS 332
- OS 333

OPEN, CLOSE, GET, PUT MACROS 333
RECORD PROCESSING 334
- Using a Work Area 335
- Processing in the Input/Output Area 336

PROGRAM 14A: CREATING A TAPE FILE 336
PROGRAM 14B: RETRIEVING A TAPE FILE 337
PROGRAM 14C: UPDATING A MASTER TAPE FILE 339

15. Sequential Disk Files **351**

DISK CHARACTERISTICS 351
- Cylinders 352
- Access Timing 353
- Track Format 353
- Count, Key, Data Organization 353
- Cyclic Check Bytes 354

TRACK CAPACITY TABLES 355
VOLUME TABLE OF CONTENTS 355
DISK LABELS AND EXTENTS 356
SEQUENTIAL DISK FILE DEFINITIONS 359
- DOS 359
- OS 359

OPEN, CLOSE, GET, PUT MACROS 359
PROCESSING SEQUENTIAL DISK FILES 360
PROGRAM 15A: CREATING VARIABLE-LENGTH RECORDS 362
PROGRAM 15B: PROCESSING VARIABLE-LENGTH RECORDS 366

16. Indexed Sequential Files 364

FILE ORGANIZATION 374
Prime Data Area 375
Track Index 375
Cylinder Index 375
Master Index 376
Overflow Areas 356
Cylinder Overflow Area 356
Independent Overflow Area 356
INDEXED SEQUENTIAL FILE DEFINITIONS 377
DOS 378
OS 379
ERROR AND STATUS BYTE 380
MACROS FOR FILE CREATION AND EXTENSION 381
DTFIS 381
DCB 381
OPEN 381
SETFL 382
WRITE or PUT 382
ENDFL 383
CLOSE 383
PROGRAM 16A: CREATING AN INVENTORY MASTER FILE 383
MACROS FOR SEQUENTIAL RETRIEVAL 387
DTFIS 387
DCB 387
OPEN and CLOSE 387
SETL 387
GET 388
PUT or PUTX 389
ESETL 390
PROGRAM 16B: INVENTORY LISTING BY SEQUENTIAL RETRIEVAL 390
MACROS FOR DIRECT RETRIEVAL AND UPDATE 390
DTFIS 394
DCB 394
READ 394
WAITF or WAIT 395
WRITE 395
PROGRAM 16C: INVENTORY MASTER DIRECT RETRIEVAL AND UPDATE 396
MACROS FOR ADDING AND DELETING RECORDS 401
DTFIS 401
DCB 402
READ or GET 402
WRITE or PUT 402
PROGRAM 16D: ADDING AND DELETING INVENTORY RECORDS 403
PROGRAM 16E: REORGANIZING THE INVENTORY FILE 403

SUMMARY OF PROCESSING MACROS, I/O AREAS, AND WORK AREAS 407

17. VSAM Files **414**

PHYSICAL ORGANIZATION 414
Control Intervals 415
Control Areas 415
LOGICAL ORGANIZATION 416
ESDS 416
KSDS 416
RRDS 417
Comparison of the Three Types 418
COMPONENTS OF VSAM DATA SETS 418
Cluster 418
Data 418
Index 418
Alternate Indexes 419
ACCESSING AND PROCESSING VSAM RECORDS 419
Accessing Records 419
Processing Records 420
Sequential Processing 420
Direct Processing 420
Skip Sequential Processing 420
ACCESS METHOD SERVICES (AMS) 421
The DEFINE Command 421
CLUSTER 421
DATA 422
INDEX 422
CONTROL BLOCK MACROS 424
ACB 424
RPL 425
EXLST 425
REQUEST MACROS 426
OPEN 426
CLOSE 426
GET 427
PUT 427
POINT 427
ERASE 428
EXECUTABLE CONTROL BLOCK MACROS 428
GENCB 428
MODCB 428
TESTCB 429
SHOWCB 429
PROGRAM 17: REORGANIZE VSAM FILE 430