

*The Motorola MC68000
Microprocessor Family:
Assembly Language,
Interface Design,
and System Design*

Thomas L. Harman

Barbara Lawson

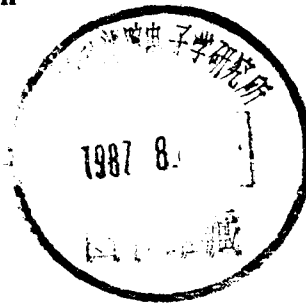


***The Motorola MC68000
Microprocessor Family:
Assembly Language,
Interface Design,
and System Design***

Thomas L. Harman

School of Sciences and Technologies
University of Houston, Clear Lake

Barbara Lawson



Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632

8750159

Library of Congress Cataloging in Publication Data

Harman, Thomas L.

The Motorola MC 68000 microprocessor family.

Includes index.

1. Motorola 68000 (Microprocessor) 2. Motorola 68000 (Microprocessor)—Programming. 3. Assembler language (Computer program language) 4. Microcomputers.

I. Lawson, Barbara. II. Title.

QA76.8.M6895H37 1985 001.64 84-11480

ISBN 0-13-603960-X

Editorial/production supervision: *Raeia Maes*

Cover design: *Photo Plus Art*

Manufacturing buyer: *Gordon Osbourne*

© 1985 by Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632

All rights reserved. No part of this book
may be reproduced, in any form or by any means,
without permission in writing from the publisher.

Printed in the United States of America

10 9 8 7 6 5 4 3 2

ISBN 0-13-603960-X 01

Prentice-Hall International, Inc., *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Prentice-Hall of Southeast Asia Pte. Ltd., *Singapore*
Whitehall Books Limited, *Wellington, New Zealand*

Preface

The introduction of the Motorola MC68000 family of microprocessors ushered in a new generation of processors. This family includes the MC68000 and the MC68010. Both are single-chip microprocessors designed to function as the central processing units of sophisticated computer systems. The characteristics and uses of these Motorola processors are treated in detail in this book. For these processors, the book covers assembly language; programming, interface design, and system design.

One important purpose of the book is to introduce the student or the practicing computer professional to all of the significant aspects of design using the MC68000. In addition, the book can serve as a reference in which topics are organized according to function and importance for the design of programs, interfaces, or systems.

This book is organized into three parts as indicated in the Chapter Descriptions below. The first four chapters present the MC68000 family to the reader and also cover other introductory material. Chapters 5 through 9 treat assembly language programming techniques for the processors. Chapters 10 through 12 are concerned with system design and development for MC68000-based computer systems. Appendices I through V summarize pertinent material about the MC68000, MC68008, and MC68010. Finally, selected answers to problems given in the chapters are included at the end of the book.

A detailed summary of the major topics covered in the book, a list of the acronyms used in the book, a list of programs by topic, and a list of additional materials that may be used in conjunction with this book are given on the following pages.

CHAPTER DESCRIPTIONS

Chapters 1 through 4 introduce the reader to the MC68000 family of products, microcomputers, fundamentals of machine arithmetic, and the MC68000 central processing unit.

- Chapter 1 presents a number of applications of the Motorola MC68000 microprocessor. The hardware and software support for the MC68000 family of integrated circuit devices is also described.
- Chapter 2 discusses the organization of typical microcomputer systems. The function of the major system components (CPU, memory, input/output) is described and the importance of the CPU word length and addressing range is presented. Three views of the system as seen by the system designer, the assembly language programmer, and the interface designer are given.
- Chapter 3 explains the internal representation of numbers and characters as used in MC68000 systems. Binary, decimal, and floating-point notations are treated along with details of arithmetic operations. The ASCII code for alphanumeric characters is also presented.
- Chapter 4 is devoted to a discussion of the characteristics of the MC68000 processor. The MC68000 is first introduced as an integrated circuit chip before its characteristics as a programmable processor are presented. The organization of memory in a typical MC68000-based system is also covered.

Chapters 5 through 9 are devoted to programming techniques using the MC68000. The assembly language for the processor is used to explain the many capabilities of the MC68000.

- Chapter 5 introduces the MC68000 assembly language. Software development, assembly language features, and the various addressing modes of the MC68000 are covered.
- Chapter 6 presents three important categories of instructions for the MC68000. Instructions for data transfer, program control, and subroutines are treated.
- Chapter 7 contains explanations and program examples concerning the arithmetic capability of the MC68000. Binary arithmetic, decimal arithmetic, and conversions between ASCII, binary, and BCD are covered.
- Chapter 8 introduces the logical instructions, shift and rotate instructions, and bit-manipulation instructions.
- Chapter 9 completes the study of basic programming techniques. Methods of creating position independent code are covered. Program examples are given for manipulation of data structures, including arrays and lists. More advanced subroutine techniques are presented.

Chapters 10, 11, and 12 are devoted to aspects of the MC68000 that determine the operation of the computer system. Chapters 10 and 11 are of concern to the system designer and the programmer who create supervisor programs. Chapter 12 covers I/O programming and details of hardware design.

- Chapter 10 considers the processor's various states and modes of operation. The assembly language instructions to control the processor and examples of initialization procedures are presented.
- Chapter 11 covers exception processing. The exceptions include interrupts, traps, and various error conditions recognized by the CPU during program execution.
- Chapter 12 presents the interfacing requirements and I/O programming for MC68000 systems.

Appendices I through V contain summary material. Appendix I contains the ASCII character codes and a table of powers of 2 and 16. Appendix II, Appendix III, and Appendix IV contain a summary of the MC68000 family characteristics, the assembly language instruction set, and the machine language instruction set for the MC68000, MC68008, and MC68010. Appendix V presents the memory map for Motorola's Design Module.

SUMMARY OF MAJOR TOPICS

General Programming

Data Types	Chapter 3
Machine Language	Section 4.5
Memory Addressing	Section 4.6
Assembly Language	Chapter 5
Summary	Appendices

MC68000 Processor

Addressing Modes	Sections 4.4, 5.3, and 9.2
Instruction Set	
Introduction	Section 4.3
Major Instructions	Chapters 6, 7, and 8
Register Set	Sections 4.2 and 10.2
Summary	Appendices

Programming Techniques

Arithmetic and Logical	
Data Types	Chapter 3
Condition Codes	Section 6.3 and 7.1
Arithmetic Operations	Chapter 7

Logical and Bit	Chapter 8
Arrays	Section 9.3
Data Structures	
Addressing	Sections 4.4, 5.3, and 9.1
Examples	Section 9.3
Position Independent Code	Section 9.2
Subroutines	Sections 6.3 and 9.4

System Development and Supervisor Programs

General	
Introduction	Section 1.2 and Chapter 2
Supervisor versus User Mode	Section 4.2
Memory Organization	Section 4.6
Test and Set (TAS)	Section 8.3
Position Independent Code	Section 9.2
System Operation	Chapter 10
Exception Processing	Chapter 11
Input and Output Programs	Section 12.2
Interrupt Techniques	
Introduction	Chapter 10
Processing	Section 11.4
Examples	Section 12.2

Hardware

Introduction	Section 4.1
Interfacing	Chapter 12

MC68008

Instruction Execution Times	Appendix IVE
-----------------------------	--------------

MC68010

Introduction	Sections 1.2 and 2.3
Virtual Memory	Sections 2.3 and 11.3
Stack Usage	Section 4.2
Status Register Access	Sections 10.1 and 10.2
RTE Instruction	Section 10.2
Vector Table Initialization	Section 10.3
Exception Processing	Chapter 11
Instruction Execution Time	Appendix IVF
Signal Lines	Section 12.4

LIST OF ACRONYMS

ACIA	Asynchronous Communications Interface Adapter
ASCII	American Standard Code for Information Interchange
ALU	Arithmetic and Logic Unit
BCD	Binary Coded Decimal
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CPU	Central Processing Unit
CRT	Cathode-Ray Tube
DIP	Dual-in-Line Package
DMA	Direct Memory Address
I/O	Input and Output
MMU	Memory Management Unit
MOS	Metal-Oxide Semiconductor
PIA	Peripheral Interface Adapter
PLA	Programmed Logic Array
PTM	Programmer Timer Module
RAM	Random Access Memory (Read/Write)
ROM	Read Only Memory
SSI	Small Scale Integration
VLSI	Very Large Scale Integration

LIST OF PROGRAMS BY TOPIC

General

Introductory Examples	Chapter 5
Conversions	
ASCII to BCD	Section 6.3
Decimal to Binary	Section 7.6
Hexadecimal to BCD or ASCII	Section 7.6
Instructions	
MOVEM	Section 6.1
EXG	Section 6.1
Bcc and DBcc	Section 6.2
Subroutine Call	Sections 6.3 and 9.4
LEA and PEA	Section 9.1
Miscellaneous	
Block Move of Data	Section 5.3
Absolute versus Relative Addressing	Section 9.2

Arithmetic

Binary Arithmetic and Logical	
Addition of Two Arrays (Vectors)	Section 6.2
Find Largest Integer in an Array	Section 6.2
Addition of Vector Elements to Compute 16- or 32-Bit Sum	Section 7.2
Sum of Differences of Two Vectors	Section 7.2
Sum of Squares of Two Vectors	Section 7.3
Average Value of N Numbers	Section 7.3
64-Bit Vector Addition	Section 7.4
32-Bit by 32-Bit Multiply	Section 7.4
Two-Line to Four-Line Decoder	Section 8.1
Multiply by Power of 2	Section 8.2
Decimal Arithmetic	
6-Digit BCD Addition	Section 7.5

Lists and Tables

Compare Byte Tables	Section 6.2
Jump Table	Section 6.2
Table Search	
Number of Negative Bytes	Section 9.1
Linked List	Section 9.3
Binary Search of a Variable-Length Table	Section 9.3
Bubble Sort	Section 9.3
Table Search	
Index into Fixed-Length Table	Section 9.3

System Operation: I/O, Interrupts, and Traps

Parity Bit Generation	Section 8.3
Memory Allocation (TAS)	Section 8.3
System Initialization Example	Section 10.3
TRAP Instruction (Call to Executive)	Section 11.1
Divide by Zero Trap	Section 11.1
Address Error Exception	Section 11.3
Serial Data Transfer (ACIA)	Section 12.2
Programmable Timer (Interrupt Processing)	Section 12.2

ADDITIONAL MATERIAL

The processor manuals provide a more complete treatment of certain characteristics of the processor than is covered in this book. The User's Manual for the MC68000 is

available from Prentice-Hall. A manual for the MC68010 can be obtained from Motorola Semiconductor Products Inc., Austin, Texas.

Since development systems vary considerably, the reader should refer to the reference manuals for the specific system on which programs are being developed. These manuals include the Assembly Language Reference Manual for the specific assembler being used, as well as the manuals that describe the use of the system, i.e., how to edit, assemble, and execute programs. Similarly, the hardware manuals for the system and the manufacturer's data sheets for specific components should be used because these documents cover details peculiar to any specific item.

ACKNOWLEDGMENTS

A number of people have made special contributions to this book. The material was developed over several semesters with many helpful suggestions for improvement from the students in the Microcomputer Design Class at the University of Houston, Clear Lake. In particular, Alice Burke, Diane Grafton, Judy Meier, and Ken Wood were very helpful.

The original manuscript was produced with the Text and Document Processor for the Hewlett Packard HP3000 computer system. Barbara Kipfer and Anna Mae Bozsini are to be thanked for their efforts in preparing the finished manuscript. Also, Tony Adams, Linda Dunn, and Mike Shelly acted as word processor operators for much of the project. Danny, Mike, Ric, and Saaid, of the University Computer Center, were also of great help in producing the final document. Others involved in the production include Becky Phillips, Dennis Pleticha, and Carol Lowry.

We wish to thank Mike Collins and James Farrell III, of Motorola, for their careful proofreading of the manuscript and other help. Diana Barber also acted as proofreader. Of course, the staff of Prentice-Hall did a fine job of putting the book into production. Our sincere appreciation goes to all these people who were involved in one way or another in the project.

Finally, our families are to be thanked for their patience and support during the long process of producing a textbook. We apologize to any other persons who helped in the endeavor but were not cited here. Please send any comments or criticisms to the authors in care of Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 07632.

*T. L. Harman
B. Lawson*

Contents

Preface ix

1 Introduction 1

1.1 Applications of the Motorola MC68000 3

1.2 The MC68000 Family 10

1.2.1 The MC68000 Processor and Related Chips 11

1.2.2 Software Support for the MC68000 Family 14

1.2.3 Development Systems 17

1.2.4 Board-Level Computer Modules 19

2 Microcomputer and Microprocessor Characteristics 21

2.1 Microcomputer Organization and Bus Structure 22

2.1.1 The Central Processing Unit and Clock 24

2.1.2 The Memory Unit 25

2.1.3 Input/Output 28

2.1.4 The Internal Bus 28

2.2 Microprocessor Word Length and Addressing Range 29

2.2.1 Word Length 29

2.2.2 Addressing Range 30

2.3 Three Views of the Microprocessor 32

2.3.1 System Design 34

- 2.3.2 Assembly Language Programming 37
- 2.3.3 Interface Design 40

3 Representation of Numbers and Characters 46

- 3.1 Number Representation 47
 - 3.1.1 Nonnegative Integers 47
 - 3.1.2 Representation of Signed Numbers 51
 - 3.1.3 Conversions between Representations 58
- 3.2 Binary-Coded Decimal 62
 - 3.2.1 BCD Representation of Positive Integers 63
 - 3.2.2 Conversion between BCD and Binary 65
 - 3.2.3 Negative BCD Integers 66
- 3.3 Floating-Point Representation 67
 - 3.3.1 Floating-Point Formats 68
 - 3.3.2 Standard Floating-Point Format 70
- 3.4 ASCII Representation of Alphanumeric Characters 72

4 Introduction to the MC68000 75

- 4.1 The MC68000 as an Integrated Circuit Processor 77
 - 4.1.1 The MC68000 as an Integrated Circuit 77
 - 4.1.2 Packaging, Signal Lines, and Power 81
 - 4.1.3 Processor Design 83
- 4.2 The MC68000 Register Set 84
 - 4.2.1 Data Registers 87
 - 4.2.2 Address Registers 88
 - 4.2.3 System Stack Pointers 90
 - 4.2.4 Program Counter 94
 - 4.2.5 Status Register 96
- 4.3 Introduction to the MC68000 Instruction Set 99
 - 4.3.1 The Clear Instruction 101
 - 4.3.2 The MOVE Instruction 103
 - 4.3.3 The ADD Instruction 105
 - 4.3.4 Other Instruction Types 105
- 4.4 Addressing Modes for the MC68000 106
 - 4.4.1 Direct Addressing 108
 - 4.4.2 Indirect Addressing 109
 - 4.4.3 Relative Addressing 110
 - 4.4.4 Immediate Addressing 110
- 4.5 Machine Language for the MC68000 112
 - 4.5.1 Single-Address Instructions 112
 - 4.5.2 Double-Address Instructions 114
- 4.6 The MC68000 and Memory Organization 117
 - 4.6.1 Memory Organization and Addressing 119
 - 4.6.2 Data Organization in Memory 119

5 MC68000 Assembly Language and Basic Instructions 123

- 5.1 Software Development 124**
 - 5.1.1 The Assembler and Listing 124
 - 5.1.2 Monitor Program 127
- 5.2 Assembly Language Characteristics 130**
 - 5.2.1 Statement Formats 130
 - 5.2.2 Assembler Directives 137
- 5.3 Addressing Modes for the MC68000 143**
 - 5.3.1 Register Direct and Absolute Addressing 144
 - 5.3.2 Indirect Addressing 148
 - 5.3.3 Predecrement and Postincrement Addressing 152
 - 5.3.4 Relative Addressing 155
 - 5.3.5 Immediate and Implied Addressing 157
 - 5.3.6 Addressing Categories 158

6 Data Transfer, Program Control, and Subroutines 161

- 6.1 Data Transfer 162**
 - 6.1.1 The MOVE Instruction 164
 - 6.1.2 Variations of MOVE 165
 - 6.1.3 Internal Data Transfer 166
- 6.2 Program Control 168**
 - 6.2.1 Unconditional Branch and Jump 169
 - 6.2.2 Branch Conditionally 171
 - 6.2.3 Branching after CMP or TST 176
 - 6.2.4 DBcc Instruction 181
- 6.3 Subroutine Usage with the MC68000 186**
 - 6.3.1 Invoking Subroutines 187
 - 6.3.2 Program Structure 189

7 Arithmetic Operations 196

- 7.1 Some Details of Binary Arithmetic 197**
- 7.2 Addition and Subtraction 200**
- 7.3 Multiplication and Division 205**
- 7.4 Multiple-Precision Arithmetic 211**
 - 7.4.1 Addition and Subtraction 213
 - 7.4.2 Multiplication 216
- 7.5 Decimal Arithmetic 219**
- 7.6 Conversions between ASCII, Binary, and BCD 223**

8 Logical and Bit Operations 228

- 8.1 Logical Operations 228**
- 8.2 Shift and Rotate Instructions 234**

77 - 802

RL

8.3	Bit-Manipulation and Flag-Setting Instructions	238
8.3.1	Bit-Manipulation Instructions	239
8.3.2	Set According to Condition Instruction	241
8.3.3	Test and Set Instruction	242
9	Programming Techniques	247
9.1	Instructions That Manipulate Addresses	248
9.1.1	Arithmetic Address Manipulation	248
9.1.2	Transfer of Addresses	251
9.2	Position-Independent Code and Base Addressing	256
9.2.1	Position-Independent Code with (PC)	257
9.2.2	Base Register Addressing	260
9.3	Data Structures	262
9.3.1	One-Dimensional Arrays	262
9.3.2	Two-Dimensional Arrays	267
9.3.3	Linked Lists	270
9.4	Subroutine Usage and Argument Passing	275
9.4.1	Passing Arguments to Subroutines	275
9.4.2	Stack Frames	280
10	System Operation	285
10.1	Processor States and Modes	287
10.1.1	Normal, Exception, and Halted States	287
10.1.2	Supervisor and User Modes	288
10.2	System Control Instructions	290
10.2.1	Status Register Modification	290
10.2.2	User Stack Pointer Manipulation	292
10.2.3	Status Register Access in User Mode	293
10.2.4	RTE Instruction	294
10.2.5	RESET Instruction	295
10.3	System Initialization	295
10.3.1	Initialization Procedure	297
10.3.2	Initialization Example	300
11	Exception Processing	302
11.1	Exceptions Caused by Program Execution	305
11.1.1	The TRAP Instruction	306
11.1.2	Divide-by-Zero Trap and TRAPV	308
11.1.3	CHK Instruction	309
11.1.4	Unimplemented Instruction Trap	311
11.2	The Trace Exception	312

- 11.3 Error Conditions Causing Traps 313**
 - 11.3.1 Privilege Violations 314
 - 11.3.2 Illegal Instruction 314
 - 11.3.3 Address Error 314
 - 11.3.4 Bus Error 316
 - 11.3.5 Halted State 316
- 11.4 Interrupt Processing by the MC68000 310**
 - 11.4.1 Interrupt Processing 319
- 11.5 Exception Priority 322**

12 Interfacing and I/O Programming 324

- 12.1 Interface Design 324**
 - 12.1.1 Functional Design of Interfaces 325
 - 12.1.2 Peripheral Chips as Interfaces 327
 - 12.1.3 MC68000 System Support Chips 329
- 12.2 I/O Programming 331**
 - 12.2.1 RESET and MOVEP Instructions 331
 - 12.2.2 I/O Transfer Techniques 332
 - 12.2.3 I/O Features of the Design Module 336
- 12.3 Instruction Execution Times 349**
- 12.4 Hardware Considerations 352**
 - 12.4.1 Data Transfer Operations 355
 - 12.4.2 Function Code Lines and Memory Usage 358
 - 12.4.3 Interrupt Processing 360
 - 12.4.4 Bus Control and Bus Error 361
 - 12.4.5 Interfacing to MC68000 Family Chips 364

Appendix I: ASCII Character Set and Powers of Two and Sixteen 369

Appendix II: MC68000 Characteristics 375

- IIA Electrical Characteristics 376
- IIB Data Organization in Memory 378
- IIC Exception Vector Assignments 380
- IID Programming Model 382

Appendix III: Assembly Language Set 383

Appendix IV: Machine Language Characteristics of the MC68000, MC68008, and MC68010 389

- IVA Condition Codes Computation 390

FVB Instruction Set Details	394
IVC MC68000 Instruction Format Summary	504
IVD Instruction Execution Times	526
IVE MC68008 Instruction Execution Times	535
IVF MC68010 Instruction Execution Times	544
IVG MC68010 LOOP Mode Operation	554
IVH MC68000 Prefetch	556

Appendix V: Design Module Memory Map 559

Answers to Selected Exercises	561
--------------------------------------	------------

Index	570
--------------	------------

1

Introduction

A decade of advances in integrated circuit technology was crowned by the announcement in 1980 of the commercial availability of the Motorola MC68000 *microprocessor*. This microprocessor and several others, which are identified as *16-bit microprocessors*, function as the central processing units (CPUs) of computers, but they have been scaled down so that the tens of thousands of circuit elements of the processor fit on a silicon chip about one-fourth of an inch (6.4 millimeters) on a side. When suitably packaged and electrically connected to other integrated circuits, which serve as memory for data and program storage and include input and output circuitry, the processor chip is the central element in a *microcomputer* system. The dimensions of the system may be 12 inches by 12 inches by 0.75 inch (30 by 30 by 1.9 centimeters) before it is enclosed in a cabinet to provide physical protection. The addition of a power supply and suitable peripheral devices such as cathode-ray tube (CRT) terminals yields a computer system capable of being programmed to accomplish a wide variety of tasks. The microprocessor itself controls the detailed operation of the system at the hardware level and performs the arithmetic, logical, or other operations required for the application involved. Figure 1.1 shows the basic hardware elements of a typical microcomputer system useful for personal or small business applications.

The excitement generated by the introduction of the MC68000 and other microprocessors of its class was not caused by the promise of small, low-cost computer systems. Such systems, based on previous microprocessors, were already commercially available. Rather, the greatly enhanced capability of these 16-bit microprocessors caused attention to be focused on the wide variety of potential applications for microcomputers. These processors definitely rival and sometimes exceed the ability of the central processing units of many minicomputers. Their speeds of operation, as an an-