

# ELEMENTS OF MICRO- PROGRAMMING

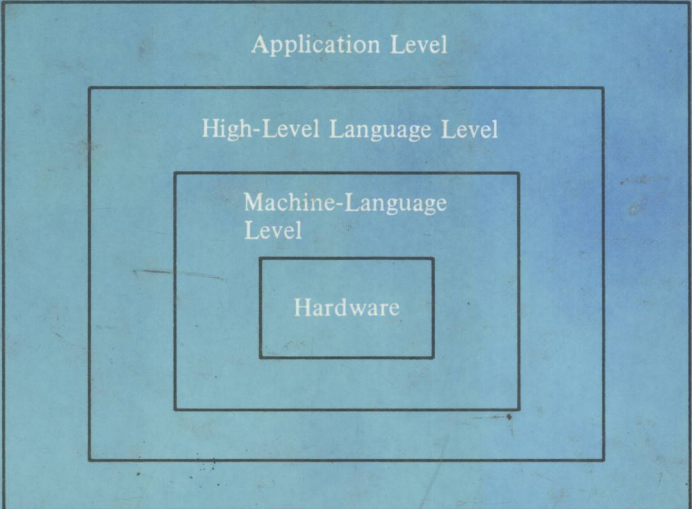
DILIP K. BANERJI  
JACQUES RAYMOND

Application Level

High-Level Language Level

Machine-Language  
Level

Hardware



TP 31  
B 29

8361244

---

# ELEMENTS OF MICROPROGRAMMING

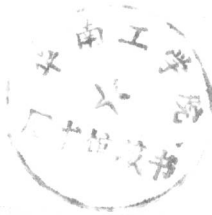
**DILIP K. BANERJI**

*Jawaharlal Nehru University  
New Delhi, India*

**JACQUES RAYMOND**

*University of Ottawa  
Ottawa, Ontario, Canada*

---



E8361244

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

533

Library of Congress Cataloging in Publication Data

BANERJI, DILIP K.  
Elements of microprogramming.  
(Prentice-Hall software series)  
Includes bibliographies and index.  
1. Microprogramming. I. Raymond, Jacques.  
II. Title. III. Series.  
QA76.6.B347 001.64'2 81-10566  
ISBN 0-13-267146-8 AACR2

© 1982 by Prentice-Hall, Inc., Englewood Cliffs, N.J. 07632

Prentice-Hall Software Series

Brian W. Kernighan, advisor

*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.*

Editorial/production supervision and interior design

by *Aliza Greenblatt*

Cover Design

by *Judith A. Matz*

Manufacturing buyer

*Gordon Osbourne*

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

**To our students**

PRENTICE-HALL INTERNATIONAL, INC., *London*  
PRENTICE-HALL OF AUSTRALIA PTY. LIMITED, *Sydney*  
PRENTICE-HALL OF CANADA, LTD., *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

---

---

This book has essentially evolved out of the course CSI 4114/ELG 5189, “Microprogramming and Machine Architecture,” that the authors have offered at the University of Ottawa since 1971. Since the material for the course is derived from various sources, including some of our own work, our students repeatedly urged us to put it all together as a package, which eventually took the form of a manuscript.

The book is intended to provide a comprehensive coverage of the basic principles, practices, and applications of microprogramming. We have endeavored to stick to this objective as much as possible. Admittedly, however, as in any book, there may be a few shortcomings. For the positive aspects of the book the authors would like to share the credit with their students and colleagues who have consistently provided encouragement and support; for any shortcomings, the buck stops at our desks.

We have deliberately avoided case studies of specific microprogrammed/microprogrammable machines. With rapidly changing computer technology, machines tend to be outdated relatively quickly, thus rendering the books containing case studies out-of-date. By concentrating on basic concepts and applications of microprogramming, we hope that this book will

remain useful for a longer period. Throughout the book we have used only one microprogrammable machine, the Microdata 1600,\* available at the University of Ottawa, to develop and illustrate microprogramming examples. Some readers may not agree with our choice of the machine, but this choice was motivated by the availability of the machine and by our desire to list tested microprograms.

Chapter 1 is purely introductory in nature; its main objective is to introduce the basic concepts of microprogramming. Chapter 2 traces the historical growth of microprogramming since the introduction of the concept in 1951 by Wilkes. The evolution of microprogramming up to the present stage as an integral part of computer architecture is discussed briefly. Chapter 3 reviews the basic concepts of computer organization. It is primarily intended as a refresher course in computer organization and can be skipped by readers familiar with computer hardware and organization. Chapter 4 discusses both the hardware resources required to support microprogramming and the basic schemes for implementing microprogram control and technologies used therein. Chapter 5 analyzes, in depth, the various aspects of microinstruction design, formats, and execution. It also discusses various design tradeoffs in microinstruction design.

Chapters 6 through 8 are concerned with microprogramming practice. In Chapter 6 we discuss the various aspects of developing microprograms. Starting with absolute coding, we discuss the need for a micro-assembly language, and then the need and availability of high-level languages for microprogramming. In Chapter 7 we are concerned with the microprogram implementation of some common machine instructions and functions. The example microprograms have been written for and tested on the Microdata 1600 machine. Chapter 8 discusses the effect of various factors on the performance of a microprogrammed system. In addition, it presents techniques for optimizing the size of a microprogram.

Chapters 9 through 13 essentially cover the various applications of microprogramming. In Chapter 9 we discuss the concept of emulation and show how this is achieved through microprogramming. A novel feature of this chapter is the section on I/O emulation which is generally not available in the existing texts. The emulation of an existing machine on the Microdata 1600 is described in some detail. Chapter 10 is concerned with microprogramming support for operating systems as well as high-level language processing. General principles of how this is achieved are discussed, along with a few specific examples of such applications. In Chapter 11, we discuss microprogrammable (bit-slice) microprocessors. The discussion is centered around the Intel 3000† microprocessor family which can be microprogrammed to suit user requirements. Chapter 12 describes the use of microprogramming to support block structured programming. This is

\*Reprinted with permission of Microdata Corporation, Irvine, California.

†Reprinted by Permission of Intel Corporation, Copyright 1976.

illustrated by an example implemented on the Microdata 1600\* machine. Finally, Chapter 13 describes some of the other applications of microprogramming: virtual memory implementation, microdiagnostics, implementation of specialized functions for computer graphics, and implementation of a communications controller. These are only a few representative examples of the other applications.

Three appendices provide useful background information to the reader. Appendix A provides a summary of the features of an emulated (target) machine discussed in Chapter 9, whereas Appendix B describes the features of the microprogrammable host machine. Appendix C provides a complete microprogram listing of the emulator discussed in Chapter 9. The References and Further Readings at the end of chapters provide a useful guide to relevant literature in the field.

The book is intended for use at Computer Science/Electrical Engineering senior and graduate levels. Chapters 1 through 8 are suitable for seniors, whereas the inclusion of Chapters 9 through 13 would make it suitable for graduate level. Some background in computer organization and some exposure to assembly language programming would be helpful in reading the book. The book can also be used as a handy reference for practicing engineers and computer professionals. It can be used either as a primary text for a full course on microprogramming or as a supplementary text in a course on computer architecture/organization. The introductory chapters can be used to supplement a basic computer organization course, while other chapters can supplement a more advanced course on architecture/organization. At the University of Ottawa it is to be used as a primary text for a full course on microprogramming.

No book can be completed without the authors receiving active support from various sources. In this context we would like to thank Data General Corporation, Microdata Corporation, and Intel Corporation for giving us permission to use some of their technical literature in the book. Many of our colleagues and students both at the University of Ottawa & JNU have helped in the preparation of the manuscript. In particular, the assistance of Peter Hickey at Ottawa in testing many of the microprograms is acknowledged. At JNU, Prasenjit Biswas, Abir Bhattacharya, and Arun Majumdar have helped a good deal. Without the assistance of these individuals, it would have been difficult to complete the book. Throughout the course of writing the book, our families have been very patient and have consistently rendered moral support. Finally, our special thanks are due to Claudette Henderson for her excellent typing.

**DILIP K. BANERJI**  
*Jawaharlal Nehru University*  
*New Delhi, India*

**JACQUES RAYMOND**  
*University of Ottawa*  
*Ottawa, Ontario, Canada*

8361244

---

---

# Contents

---

---

*Preface*

xv



## PART ONE REVIEW

*Chapter One Basics of Microprogramming* 3

- 1.1 Introduction 3
- 1.2 The Control Unit 4
- 1.3 Concepts of Microprogramming 9
- 1.4 Advantages and Disadvantages of Microprogramming 12
- Summary 15
- References 16
- Further Readings 16

**Chapter Two Historical Evolution of Microprogramming 17**

- 2.1 Introduction 17
- 2.2 History of Growth 19
- 2.3 User-Microprogrammable Machines 20
- 2.4 History of Applications 21
- Summary 22
- References 23
- Further Readings 23

**Chapter Three Basic Computer Organization: an Overview 25**

- 3.1 Introduction 25
- 3.2 CPU Organization 26
  - 3.2.1 Index Registers 28; 3.2.2 Base Register 29;
  - 3.2.3 General-Purpose Registers 30;
  - 3.2.4 Input/Output Register 32;
  - 3.2.5 Return Address Stack (RAS) 33
- 3.3 Instruction Formats 33
- 3.4 Control Flip-Flops and Their Functions 36
  - 3.4.1 Fetching and Execution of Instructions 37;
  - 3.4.2 Program State Flip-Flops 41;
  - 3.4.3 Interrupt Mechanism and Flip-Flop 42;
  - 3.4.4 I/O Interrupt Handling 44;
  - 3.4.5 Direct Memory Access (DMA) and DMA Flip-Flop 47;
  - 3.4.6 Traps and Trap Flip-Flop 50;
  - 3.4.7 Programmed Interrupt 51; 3.4.8 Hard Errors 51
- 3.5 Instruction Classification 52
  - 3.5.1 Arithmetic Instructions 53; 3.5.2 Logical Instructions 53;
  - 3.5.3 Shift/Rotate Instructions 53;
  - 3.5.4 Comparison Instructions 55; 3.5.5 Branch Instructions 55;
  - 3.5.6 Subroutine Linkage Instructions 56;
  - 3.5.7 Data Movement Instructions 56;
  - 3.5.8 I/O and Privileged Group of Instructions 57;
  - 3.5.9 Control Group Instructions 59;
  - 3.5.10 Miscellaneous Instructions 59
- 3.6 A Proposed Microinstruction Format 60
  - Summary 61
  - References 62



## PART TWO      MICROPROGRAMMING HARDWARE AND MICROINSTRUCTIONS

### *Chapter Four    Hardware for Microprogramming*

65

- 4.1 Microprogramming Control Unit 65
  - 4.1.1 *Functional Units* 65;
  - 4.1.2 *Control Memory Sequencing Schemes* 67;
  - 4.1.3 *Other Hardware Facilities* 70
- 4.2 Review of Control Memory Technology 71
  - 4.2.1 *Read/Write Memories* 71;
  - 4.2.2 *Read-Only Memories (ROM)* 72; *Future Outlook* 77
- 4.3 Levels of Control 78
  - 4.3.1 *Concepts of Levels* 78;
  - 4.3.2 *The Machine-Language Level* 80;
  - 4.3.3 *The Microprogram Level* 80; 4.3.4 *Further Levels* 81
- References 83
- Further Readings 83
- Exercises 85



### *Chapter Five    Microinstructions*

86

- Introduction 86
- 5.1 Microinstruction Structure 87
  - 5.1.1 *Horizontal Microinstructions* 87;
  - 5.1.2 *Vertical Microinstructions* 88;
  - 5.1.3 *Diagonal Microinstructions* 89;
  - 5.1.4 *Encoding of Microinstruction Fields* 92
- 5.2 Microinstruction Execution 93
  - 5.2.1 *Basic Microinstruction Cycle* 93;
  - 5.2.2 *Vertical Microinstruction Execution Example* 96;
  - 5.2.3 *Horizontal Microinstruction Execution Example* 97;
  - 5.2.4 *Diagonal Microinstruction Execution Example* 97
- 5.3 Microinstruction Design: Some Considerations 98
  - 5.3.1 *Vertical versus Nonvertical Format* 98;
  - 5.3.2 *Minimization of Microinstruction Size* 100;
  - 5.3.3 *Other Considerations* 105
- References 107
- Exercises 109

**PART THREE      MICROPROGRAMMING PRACTICE**

***Chapter Six    Developing a Microprogram*** **113**

- 6.1 Direct Microprogramming 113
  - 6.1.1 *Writing Absolute Code* 113;
  - 6.1.2 *Testing a Microprogram* 115;
  - 6.1.3 *Obtaining the Firmware* 119
- 6.2 Assemblers 120
  - 6.2.1 *Microassembler Language* 120;
  - 6.2.2 *The Assembly Process* 122;
  - 6.2.3 *Extra Features of Assemblers* 123
- 6.3 High Level Microprogramming Languages 125
  - 6.3.1 *Necessity and Advantages* 125;
  - 6.3.2 *Disadvantages and Trade-Offs* 126;
  - 6.3.3 *Considerations in the Design of a Microprogramming Language* 127;
  - 6.3.4 *High Level Languages for Microprogramming* 128
- 6.4 Micro Operating Systems 131
  - 6.4.1 *Introduction* 131; 6.4.2 *Components* 131
  - References 132
  - Further Readings 132
  - Exercises 136

***Chapter Seven    Implementation of Some Common Machine Instructions and Functions*** **137**

- 7.1 Introduction 137
- 7.2 Some Common Machine Instructions 141
- 7.3 Specialized Instructions 146
- 7.4 Examples of I/O Instructions 156
  - 7.4.1 *Microdata 1600 I/O System* 156; 7.4.2 *Examples* 158
- 7.5 Interrupt Handling 162
  - 7.5.1 *Microdata 1600 Interrupt System* 163;
  - 7.5.2 *Examples of Interrupt Handling* 166
- 7.6 Stack Processing 169
  - References 173
  - Further Readings 173
  - Exercises 174

**Chapter Eight Performance and Optimization**

175

- 8.1 Introduction 175
- 8.2 Control Store Optimization 176
  - 8.2.1 *Use of Parallelism* 176;
  - 8.2.2 *Dynamic Address Modification* 177;
  - 8.2.3 *Use of Subroutines* 179; 8.2.4 *Common Segments* 180;
  - 8.2.5 *Extending the Control Store* 180;
  - 8.2.6 *Formal Techniques* 184
- 8.3 Execution Time Optimization 190
  - 8.3.1 *Use of Parallelism* 190;
  - 8.3.2 *Use of Suitable Programming Methodology* 192;
  - 8.3.3 *Algorithms* 193
- References 195
- Further Readings 195
- Exercises 196

**PART FOUR MICROPROGRAMMING APPLICATIONS****Chapter Nine Microprogramming and Emulation**

199

- 9.1 Basic Concepts 199
  - 9.1.1 *Introduction* 199;
  - 9.1.2 *Characteristics of Host and Target Machines* 200;
  - 9.1.3 *Strategy for Emulation* 201
- 9.2 CPU Emulation 202
  - 9.2.1 *Data Path Structure* 202;
  - 9.2.2 *Functional Processing Unit* 203;
  - 9.2.3 *Mapping of Registers, Local Storage, and Main Memory* 205;
  - 9.2.4 *Example of CPU Emulation* 206
- 9.3 Emulation of Input/Output Systems 213
  - 9.3.1 *I/O Device Mapping* 216;
  - 9.3.2 *Device Operational Characteristics* 216;
  - 9.3.3 *Data Formatting* 217; 9.3.4 *I/O Bus Structure* 217;
  - 9.3.5 *I/O Interrupt Structures* 217;
  - 9.3.6 *Timing Considerations* 217

- 9.4 Emulation of Interrupt Systems 218
  - 9.4.1 *Mapping of Interrupt Systems* 218;
  - 9.4.2 *Interrupt Handling* 219;
  - 9.4.3 *Example: Emulation of Nova I/O System on the Microdata 1600* 220
- 9.5 General Remarks 232
  - 9.5.1 *Hardware, Firmware, Software Trade-offs in Emulation* 232;
  - 9.5.2 *Universal Host Machine* 233;
  - 9.5.3 *Emulation of Front Panel Operation* 234
- References 235
- Further Readings 235
- Exercises 237

**Chapter Ten Microprogramming Support for Operating Systems and High Level Languages**

238

- 10.1 Operating System Support 239
  - 10.1.1 *Review of Operating System Concepts* 239;
  - 10.1.2 *OS Overhead* 241;
  - 10.1.3 *Improvements Expected From Firmware Support* 241;
  - 10.1.4 *General Considerations in Firmware Support* 242
- 10.2 Implementation 244
  - 10.2.1 *Identification of Primitives* 244;
  - 10.2.2 *Criteria For Firmware Implementation* 246;
  - 10.2.3 *Selection of Primitives for Firmware Implementation* 246;
  - 10.2.4 *Firmware Implementation* 247;
  - 10.2.5 *Firmware Support* 248
- 10.3 Examples 248
  - 10.3.1 *Microprogramming of a Scheduler* 248;
  - 10.3.2 *Microprogramming Some Basic Operations* 250;
  - 10.3.3 *Other Examples* 252
- 10.4 Firmware Support for High-Level Languages 254
  - 10.4.1 *The Problem* 254;
  - 10.4.2 *An Overview of the Work Done* 257
- References 264
- Further Readings 265
- Exercises 267

**PART FIVE    EXAMPLES**

<b><i>Chapter Eleven    Microprogrammable Microprocessors:                           An Example</i></b>	<b>271</b>
11.1 Introduction	271
11.2 The Intel Series 3000	275
11.2.1 3001 Microprogram Control Unit	277;
11.2.2 Functional Description of the MCU	280;
11.2.3 Extended Addressing	287;
11.2.4 3002 Central Processing Element (CPE)	288;
11.2.5 Functional Description of the 3002 CPE	290
11.3 Cross Microprogramming System (CROMIS)	297
11.3.1 XMAS Language	298;
11.3.2 XMAP Language and Control Language	312;
11.3.3 Microprogramming Techniques for Series 3000	313
11.4 Additional Support: In-Circuit Emulation	321
References	323
<b><i>Chapter Twelve    A Firmware Implementation                           of Block-Structured Programming</i></b>	<b>324</b>
12.1 Top-down Structured Programming at Machine Level (TDSP)	325
12.1.1 Primitives and Control Programs	325;
12.1.2 The Hierarchy of Control	327;
12.1.3 Arguments	328;
12.1.4 Machine Instructions for Control	329;
12.1.5 Overview of the Execution of One Instruction	329
12.2 Description of the Control Program Instructions	331
12.2.1 Writing a Control Program	331;
12.2.2 Description of the Execution of a Control Instruction	338
References and Further Readings	345
Exercises	346
<b><i>Chapter Thirteen    Other Examples of Applications</i></b>	<b>347</b>
13.1 Virtual Memory Implementation Through Microprogramming	347
13.1.1 Concept of Virtual Memory	348;
13.1.2 An Example of DAT Implementation	348

13.2	Microdiagnostics	355
	13.2.1	<i>Introduction</i> 355;
	13.2.2	<i>Advantages of Microdiagnostics</i> 356;
	13.2.3	<i>Microdiagnostics Implementation</i> 357
13.3	Firmware Implementation of Some Functions of a Graphics Terminal	358
	13.3.1	<i>Introduction</i> 358;
	13.3.2	<i>A Microprogrammed Graphics Terminal</i> 359;
	13.3.3	<i>Architecture of the Graphics Terminal</i> 361;
	13.3.4	<i>Operation of the Terminal</i> 362
13.4	A Communications Controller	362
	13.4.1	<i>Introduction</i> 362;
	13.4.2	<i>Example of a Microprogrammed Controller</i> 363
	References	376
	Further Readings	376
	Exercises	378
 <i>Appendix A The Nova Computer</i>		 379
 <i>Appendix B The Microdata 1600</i>		 396
 <i>Appendix C Nova Emulator</i>		 414
 <i>Index</i>		 431

---

---

# **PART ONE**

## **Review**

---

---





---

---

# CHAPTER ONE

## Basics of Microprogramming

---

---

### 1.1 INTRODUCTION

Since the introduction of the concept in 1951 by Wilkes [1], microprogramming has attained a great deal of acceptance and significance, not only as an alternative method of control unit design in a computer\*, but also as a powerful tool in the hands of designers and users alike. The main reason for using this technique in computer control implementation is the tremendous flexibility attainable at a relatively small cost, which has made user-restructuring of computer architecture a reality. Also as Husson [2] has observed, “microprogramming has made it economically feasible to have the same comprehensive instruction set built into a whole line of new computers, even the smallest ones. Thus we now have computers that are architecturally compatible, yet their internal hardware, organization, and structure are drastically different.” Besides offering the so-called upwards and downwards

\*Although we talk of the control of a computer, the concept applies to any digital controller.