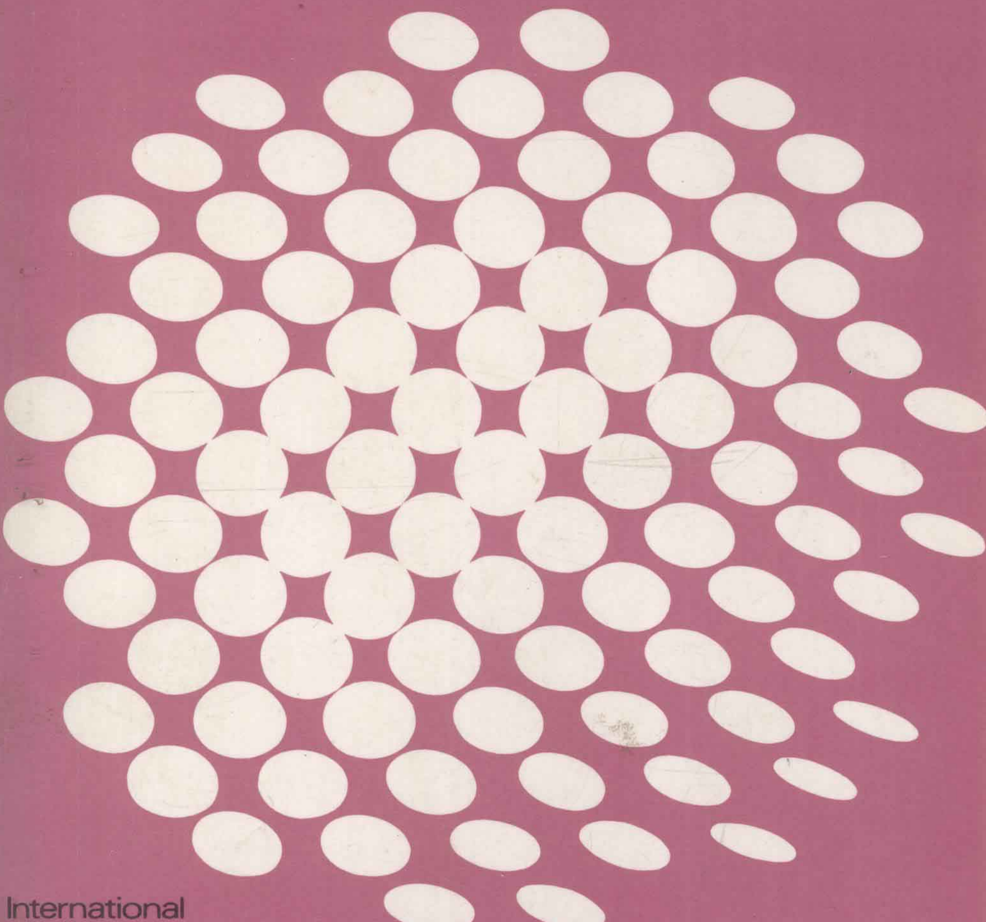
COMPUTER ORGANIZATION

V. C. HAMACHER, Z. G. VRANESIC, AND S. G. ZAKY



International
Student
Edition

COMPUTER ORGANIZATION

V. Carl Hamacher
Associate Professor of Electrical Engineering and Computer Science
University of Toronto

Zvonko G. Vranesic
Associate Professor of Electrical Engineering and Computer Science
University of Toronto

Safwat G. Zaky
Associate Professor of Electrical Engineering
University of Toronto

INTERNATIONAL STUDENT EDITION

McGRAW-HILL KOGAKUSHA, LTD.

Tokyo Auckland Bogota Guatemala Hamburg Johannesburg
Lisbon London Madrid Mexico New Delhi Panama
Paris San Juan São Paulo Singapore Sydney

COMPUTER ORGANIZATION

INTERNATIONAL STUDENT EDITION

Library of Congress Cataloging in Publication Data

Hamacher, V C
Computer organization

(McGraw-Hill computer science series)
Includes bibliographies and index.
1. Computer architecture. I. Vranesic, Zvonko G., joint author. II. Zaky, Safwat G., joint author. III. Title.
QA76.9.A73H35 621.3819'52 77-11051
ISBN 0-07-025681-0

This book was set in Times Roman by Black Dot, Inc.
The editors were Charles E. Stewart and Barbara Brooks;
the production supervisor was Dennis J. Conroy.
The drawings were done by Oxford Illustrators Limited.

KOSAIDO PRINTING CO., LTD. TOKYO, JAPAN

COMPUTER ORGANIZATION

McGraw-Hill Computer Science Series

Edward A. Feigenbaum
Stanford University

Richard W. Hamming
Naval Post Graduate School

Allen *Anatomy of LISP*
Bell and Newell *Computer Structures: Readings and Examples*
Donovan *Systems Programming*
Gear *Computer Organization and Programming*
Givone *Introduction to Switching Circuit Theory*
Goodman and Hedetniemi *Introduction to the Design and Analysis of Algorithms*
Hamacher, Vranesic, and Zaky *Computer Organization*
Hamming *Introduction to Applied Numerical Analysis*
Hayes *Computer Architecture and Organization*
Hellerman *Digital Computer System Principles*
Hellerman and Conroy *Computer System Performance*
Kain *Automata Theory: Machines and Languages*
Katzan *Microprogramming Primer*
Kohavi *Switching and Finite Automata Theory*
Liu *Elements of Discrete Mathematics*
Liu *Introduction to Combinatorial Mathematics*
Madnick and Donovan *Operating Systems*
Manna *Mathematical Theory of Computation*
Newman and Sproull *Principles of Interactive Computer Graphics*
Nilsson *Problem-solving Methods in Artificial Intelligence*
Rosen *Programming Systems and Languages*
Salton *Automatic Information Organization and Retrieval*
Stone *Introduction to Computer Organization and Data Structures*
Stone and Siewiorek *Introduction to Computer Organization and Data Structures: PDP-11 Edition*
Tonge and Feldman *Computing: An Introduction to Procedures and Procedure-Followers*
Tremblay and Manohar *Discrete Mathematical Structures with Applications to Computer Science*
Tremblay and Sorenson *An Introduction to Data Structures with Applications*
Tucker *Programming Languages*
Watson *Timesharing System Design Concepts*
Wegner *Programming Languages, Information Structures, and Machine Organization*
Wiederhold *Database Design*
Winston *The Psychology of Computer Vision*

To Liz, Anne, and Shirley

PREFACE

A considerable number of books on "computer organization" aimed at first-level specialist courses in computer science and electrical engineering curricula have been published in the last 15 years. These books usually assume some basic knowledge of computer programming but may or may not assume a background in digital logic. Many books start with a description of the basic parts of digital systems and then proceed to a design of a hypothetical computer, which is intended to illustrate most concepts discussed in the book. Since such hypothetical machines are never manufactured, they are seldom subjected to practical constraints. This leaves the reader in a dilemma about the relative merits of various features.

Our resolve to write a book stems from our experience in teaching computer organization to three distinct types of undergraduates: computer science specialists, electrical engineering undergraduates, and engineering science undergraduates. We feel that a book has to come to grips with real problems, even if they are ill-defined. Thus one of our basic decisions was to avoid hypothetical machines and instead make use of examples taken from commercially available computers.

Second, we feel that it is important to recognize that digital system design is not a straightforward process of applying "optimal design" algorithms. Many design decisions are based largely on heuristic judgement and tend to be a compromise between extreme alternatives. Thus it is our goal to convey these notions to the reader.

Third, we have endeavored to provide sufficient details to force the student to dig beyond the surface when dealing with ideas that are seemingly intuitively obvious. We believe that this is best accomplished by giving real examples that are adequately documented. Block diagrams are a powerful means of describing organizational features of a computer. However, they can easily lead to an oversimplified view of the problems involved.

In view of our desire to use real machines for illustrative purposes, we felt it desirable to use one representative computer whenever possible. Our choice was the PDP-11 minicomputer. One of the main reasons for choosing a minicomputer is its manageable size and complexity. Of course, this means that some topics have to be illustrated by means of other machines. But, in our opinion, this is not a serious drawback and cannot be avoided if real machines are to be used. We are aware of a possible trap of overemphasizing minicomputers, but we feel that this will not be a serious problem.

The book is aimed at a *one-semester* course in computer science or electrical engineering programs. It assumes some very basic knowledge of programming and logic design. The latter is not necessarily a prerequisite, as we have included an extensive appendix covering the required ideas. The book will be suitable for both hardware- and software-oriented students. There is a greater emphasis on hardware, since we feel that this is the way computer organization should be taught. It is a mistake to describe computer structures solely through the eyes of a programmer, particularly for people who work with systems that involve a variety of equipment, interfacing, and communications facilities. However, although the emphasis is on computer hardware, we have addressed a number of software issues and discussed representative instances of software-hardware trade-offs in the implementation of various components of a computing system.

Let us review the topics covered in sequence, chapter by chapter. The first nine chapters constitute the primary material on computer organization as such. The last three chapters discuss system software, microprocessors, and computer communications.

Chapter 1 overviews computer structure and informally introduces a number of terms that are dealt with in more depth in the remainder of the book. A discussion is included of the basic ways that the standard functional units can be interconnected to form a complete computing system.

Chapter 2 gives a methodical treatment of addressing techniques and instruction sequencing. The PDP-11 minicomputer is used to illustrate the basic concepts. Numerous programs and program segments at the machine instruction level are used to discuss loops, subroutines, and simple input-output programming.

Chapter 3 continues the discussion of instruction sets begun in Chapter 2 and focuses on some of the problems encountered because of the "bit-space" limitations of short word-length machines. Instruction sets in computers other than the PDP-11, namely, the IBM S360/370 line and the HP3000, are also discussed. They illustrate the possibilities afforded by longer word length and stack-oriented design, respectively. Some of the organizational limitations of the very short word length microcomputer class are briefly introduced.

Chapter 4 is the beginning of a six-chapter sequence that can be considered the core of the text. These chapters treat the Central Processing Unit, Microprogrammed Control, I/O Organization, Arithmetic, Memories, and Peripherals. Chapter 4 begins with a register-transfer level treatment of the implementation of instruction fetching and execution. The constraints imposed by various CPU busing arrangements are explained, followed by a discussion of both hardwired and microprogrammed control in the CPU.

Chapter 5 extends the discussion of microprogrammed control in the CPU. The alternatives of fully decoded command words and partially encoded command words are treated, followed by a rather detailed analysis of the "next-address" generation problem in microprogram sequencing.

Input-output organization is developed in Chapter 6. The basics of I/O data transfer synchronization are presented, and then a series of increasing complexity I/O structures is explained. First, DMA methods and interrupts are introduced, and then these ideas are extended to a discussion of channels.

Chapter 7 treats the arithmetic unit of the computer. It concentrates on fixed-point add, subtract, multiply, and divide hardware, operating on 2's-complement numbers. Lookahead adders and high-speed array multipliers are discussed. The fundamentals of floating-point representation and processing are then presented.

Semiconductor and magnetic-core memories are discussed in Chapter 8. Multiple-module memory systems and caches are explained as ways of increasing main-memory bandwidth. Various cache-mapping methods are discussed. A brief introduction to virtual-memory systems is given.

A variety of peripheral devices are discussed in Chapter 9. Teletypewriters, CRT terminals, and graphic displays are analyzed in detail. This is followed by a quantitative specification of the capacities and speeds of magnetic disk, drum, and tape mass storage devices.

This completes the core chapters of the text. Most of this material should be covered if the student is to gain a balanced viewpoint of the way modern computing system components are constructed and interconnected.

The final three chapters can be used as time and class interest dictate. We feel that graduates in electrical engineering or computer science who have studied digital systems and computer organization can be expected to know something about the rudiments of computer communications and microprocessor systems. These two aspects of digital systems are becoming increasingly important. So, although we say that Chapters 11 and 12 are not core chapters, we strongly recommend that electrical engineering computer majors should be exposed to the subjects. The same is true of Chapter 10, which gives an introduction to the subject of operating-system software. In large computer systems, total performance is dependent on the ease with which the computer hardware can be dynamically shared among a number of users. This sharing is controlled by the operating-system programs. Therefore a study of computer organization should include a study of this control software, if the student is to appreciate the relative merits of various possible I/O structures.

All the material in this book can be covered in a 12-to-15-week, 3-lecture-hours-per-week course. However, as well as being suited for the usual undergraduate class teaching environment, we feel that the material is appropriate for self-study by graduates who have not specialized in computing but who have taken introductory courses or have work experience in the area. The use of real (commercially available) computers in our examples should make the book attractive to the latter readership. Abbreviated uses of the text are also possible. We have already mentioned that Chapters 10, 11, and 12 are rather specialized and can be omitted from introductory courses. Since the basics of microprogrammed control are discussed in Chapter 4, Chapter 5 can be omitted if time is a problem. Also, Chapter 3 is not essential in that its main theme is alternatives in instruction sets. Therefore, a course with 24 to 26 lectures can achieve a reasonable level of completeness based on Chapters 1, 2, 4, 6, 7, 8, and 9, as well as the logic appendix if the students do not possess this knowledge already.

The authors wish to express their thanks to all the people who have helped during the preparation of the manuscript. We are especially grateful for the detailed, constructive criticism of the manuscript by Professor Harold S. Stone. Professors Martha Sloan, Irving H. Thomae, and Marvin C. Woodfill also provided many helpful comments and suggestions. Their careful review work led to substantial improvement in many aspects of the material. Our colleague, Professor Ron Baecker, also read the entire manuscript and gave us a number of useful comments. Finally, we wish to acknowledge the efforts of Amelia Chung and Mary-Jean Clements, who typed and retyped most of the manuscript several times.

V. Carl Hamacher
Zvonko G. Vranesic
Safwat G. Zaky

CONTENTS

Preface xi

1 Basic Structure of Computers 1

1.1 Functional Units 1
1.2 Basic Operational Concepts 11
1.3 Bus Structures 13
1.4 Concluding Remarks 16

2 Addressing Methods and Machine Program Sequencing 17

2.1 Memory Locations, Addresses, and Encoding of Information 17
2.2 Main Memory Operations 20
2.3 Instruction Formats and Instruction Sequencing 21
2.4 Addressing Modes 27
2.5 The PDP-11 Addressing Modes and Instructions 33
2.6 Simple Input-Output Programming 51
2.7 Pushdown Stacks 56
2.8 Subroutines 60
2.9 Concluding Remarks 67
2.10 Problems 68

3 Instruction Sets and Their Implementation 74

3.1 The Instruction Set of the PDP-11 Minicomputer 75
3.2 Instruction Sets in Large Computers 78
3.3 Stack Computers 82
3.4 Instruction Sets in Microcomputers 93
3.5 Concluding Remarks 94
3.6 Problems 95

4 The Central Processing Unit 96

4.1 Some Fundamental Concepts 96
4.2 Execution of a Complete Instruction 107
4.3 Sequencing of Control Signals 111
4.4 Concluding Remarks 120
4.5 Problems 120

5 Microprogrammed Control 124

5.1 Microinstructions 125
5.2 Grouping of Control Signals 126
5.3 Microprogram Sequencing 130
5.4 Microinstructions with Next-Address Field 138
5.5 Prefetching of Microinstructions 141
5.6 Emulation 144
5.7 Concluding Remarks 145
5.8 Problems 146

6 Input-Output Organization 149

6.1 Addressing of I/O Devices 150
6.2 Data Transfer 151
6.3 Synchronization 155
6.4 Interrupt Handling 158
6.5 I/O Channels 165
6.6 Concluding Remarks 176
6.7 Problems 176

7 Arithmetic 181

7.1 Number Representations 182
7.2 Addition of Positive Numbers 183
7.3 Logic Design for Fast Adders 186
7.4 Addition and Subtraction of Positive and Negative Numbers 189
7.5 Overflow in Integer Arithmetic 193
7.6 Multiplication of Positive Numbers 194
7.7 Signed-Operand Multiplication 199
7.8 Fast Multiplication 202
7.9 Integer Division 208
7.10 Floating-Point Numbers and Operations 211
7.11 Concluding Remarks 220
7.12 Problems 220

8 The Main Memory 227

8.1 Some Basic Concepts 227
8.2 Semiconductor RAM Memories 231
8.3 Semiconductor ROM Memories 236
8.4 Magnetic-Core RAM Units 237
8.5 Multiple-Module Memories and Interleaving 243
8.6 Cache Memories 245
8.7 Virtual Memories 252

8.8 Concluding Remarks 254
8.9 Problems 255

9 Computer Peripherals and Interfacing 259

9.1 I/O Interfaces 260
9.2 I/O Devices 275
9.3 On-Line Storage 286
9.4 Concluding Remarks 295
9.5 Problems 296
9.6 References 298

10 Software 299

10.1 Languages and Translators 300
10.2 Loaders 302
10.3 Linkers 306
10.4 Operating Systems 310
10.5 Concluding Remarks 322
10.6 Problems 323
10.7 References 324

11 Microprocessors 325

11.1 Single-Chip Processing Elements 325
11.2 Families of Microprocessor Chips 326
11.3 Microprocessor Instruction Sets 328
11.4 Processor Bit-Slice Integrated Circuits 345
11.5 Applications of Microprocessors 347
11.6 Concluding Remarks 349
11.7 Problems 349
11.8 References 350

12 Computer Communications 351

12.1 Communication with a Remote Terminal 352
12.2 Error Control 361
12.3 Multiterminal Configurations 366
12.4 Circuit and Message Switching 379
12.5 Concluding Remarks 384
12.6 Problems 384
12.7 References 386

Bibliography 388

Appendix 390

A Logic Circuits 390

A.1 Basic Logic Functions 391
A.2 Synthesis of Logic Functions Using AND, OR, and NOT Gates 394
A.3 Minimization of Logic Expressions 396
A.4 Synthesis with NAND and NOR Gates 404

A.5 Practical Implementation of Logic Gates 406
A.6 Flip-Flops 418
A.7 Registers 428
A.8 Shift Registers 429
A.9 Counters 430
A.10 Decoders 431
A.11 Multiplexers 432
A.12 Programmable Logic Arrays (PLAs) 435
A.13 Concluding Remarks 437
A.14 Problems 439
A.15 References 442

B Instruction Set for PDP-11 Minicomputers 443

C Character Codes and Number Conversion 451

C.1 Character Codes 451
C.2 Decimal to Binary Conversion 454

Index 457

CHAPTER
ONE
BASIC STRUCTURE OF COMPUTERS

The objective of this chapter is to introduce some basic concepts and associated terminology or jargon. We will give only a broad overview of the fundamental characteristics of computers, leaving the more detailed (and precise) discussion to the subsequent chapters.

Let us first define the meaning of the word "digital computer" or simply "computer," which is often misunderstood, despite the fact that most people take it for granted. In its simplest form, a contemporary computer is a fast electronic calculating machine, which accepts digitized "input" information, processes it according to a "program" stored in its "memory," and produces the resultant "output" information.

1.1 FUNCTIONAL UNITS

The word *computer* encompasses a large variety of machines, widely differing in size, speed, and cost. It is fashionable to use more specific words to represent some subclasses of computers. Smaller machines are usually called *minicomputers*, which is a reflection on their relatively lower cost, size, and computing power. In the early 1970s the term *microcomputer* was coined to describe a very small computer, low in price, and consisting of only a few large-scale integrated (LSI) circuit packages.

Large computers are quite different from minicomputers and microcomputers in size, processing power, cost, and the complexity and sophistication of their design. Yet the basic concepts are essentially the same for all classes of computers, relying on a few well-defined ideas which we will attempt to explain. Thus the following discussion should be applicable to most general-purpose digital computers.

A computer consists of five functionally independent main parts: input, memory, arithmetic and logic, output, and control units, as indicated in Figure 1.1. The input unit accepts coded information from the outside world, either from human operators or from electromechanical devices. The information is either stored in the memory for later reference or immediately handled by the arithmetic and logic circuitry, which performs the desired operations. The processing steps are determined by a "program" stored in the memory. Finally, the results are sent back to the outside world through the output unit. All these actions are coordinated by the control unit. The diagram in Figure 1.1 does not show the connections between the various functional units. Of course, such connections must exist. However, there are a number of ways in which the connections can be made, and they will be discussed in many places throughout the book.

It is customary to refer to the arithmetic and logic circuits in conjunction with the main control circuits as the *central processing unit* (CPU). Similarly, input and output equipment is combined under the term *input-output unit* (I/O). This is reasonable in view of the fact that some standard equipment provides both input and output functions. The simplest such example is the often encountered teletypewriter terminal. We must emphasize that input and output functions are separated within the terminal. Thus the computer sees two distinct devices, even though the human operator associates them as being part of the same unit.

In large computers the main functional units may comprise a number of separate, and often sizeable, physical parts. Figure 1.2 is a photograph of such a computer. Minicomputers are much smaller in size. A basic minicomputer is often of desktop dimensions, as illustrated by the two machines in Figure 1.3.

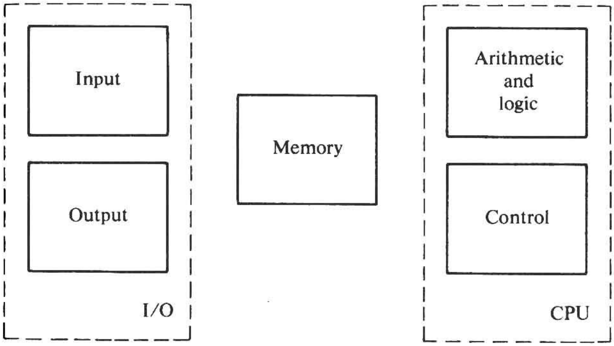


Figure 1.1 Basic functional units of a computer.

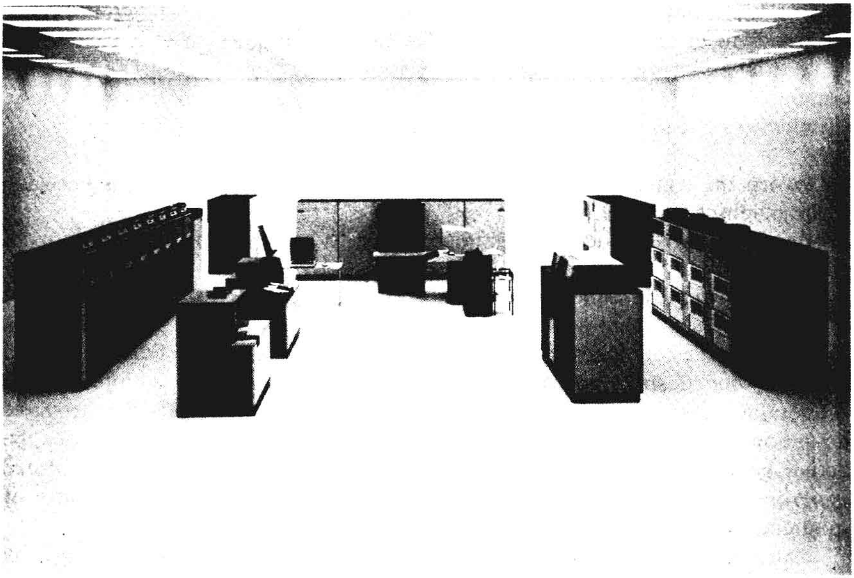


Figure 1.2 A typical large computer—IBM S370/158. *(IBM Corp. Ltd.)*

Figure 1.3 Two minicomputers—PDP8/M and PDP11/05. *(Digital Equipment Corp.)*