COMPUTER HARDWARE AND ORGANIZATION

Second Edition

M. E. SLOAN

COMPUTER HARDWARE AND ORGANIZATION
An Introduction

SECOND EDITION

M. E. Sloan
Michigan Technological University

SRA ®
SCIENCE RESEARCH ASSOCIATES, INC.
Chicago, Palo Alto, Toronto
Henley-on-Thames, Sydney
A Subsidiary of IBM

We wish to acknowledge the following for permission to reprint material:

Figures 11.5 and 11.6: From *Hewlett-Packard Journal*, June 1972. © 1972. Figures 11.7 and 11.8: From *Hewlett-Packard Journal*, March 1980. © 1980, Hewlett-Packard Company. Used with permission. Figures 11.9 and 11.10. From *Intel MCS-40 Users Manual for Logic Designers*, © 1975, Intel Corporation. Used with permission. Figures 11.11, 11.16, 11.17, 11.18, and 11.19: From *PDP8/e, PDP8/m, and PDP8/f Small Computer Handbook*, © 1973, Digital Equipment Corporation. Used with permission. Figures 11.20, 11.21, 11.22, and 11.23 courtesy of the Motorola Corporation.

Acquisition Editor	Alan W. Lowe
Project Editor	Ann Wood
Designer	Judith Olson
Cover Designer	House of Graphics
Compositor	Computer Typesetting Services, Inc.

Printed in the United States of America

Library of Congress Cataloging in Publication Data

Sloan, M. E. (Martha E.)
Computer hardware and organization.

Includes bibliographical references and index.
1. Computers. I. Title.
TK7885.S56 1983 621.3819'5 82-24116
ISBN 0-574-21425-9

10 9 8 7 6 5 4 3 2 1

PREFACE

Many changes have taken place in computers since the first edition of this book was written several years ago. In the early 1970s microprocessors were new on the scene; now they are standard. The PDP-8 and the PDP-11 were the most common computers for a first course in computer organization; now the PDP-8 has been displaced by microprocessors. In addition, the pervasiveness of computers in modern society has greatly increased the number and variety of students interested in a course in elementary computer hardware and organization. Whereas in the mid-1970s such a course was taken by computer science and engineering majors with a sprinkling of other electrical engineering students, now more than half the students typically come from a wide variety of other disciplines, including science, engineering, technology, and business. These changes and the enthusiastic acceptance of the first edition by instructors at a wide variety of schools have prompted the new emphases in this edition.

This book is intended for use in an introductory course in computer hardware and organization. It is suited for college freshmen or sophomores but, with the optional sections, is also appropriate for juniors and seniors. Students are assumed to have taken at least two years of high school mathematics. This minimal prerequisite makes the book especially suited to the wider audience for today's course; in particular, students do not need a background in electric circuits. Previous acquaintance with programming in a high-level language such as FORTRAN or Pascal is helpful but not necessary.

In this second edition the PDP-8 has been replaced as a running example by the PDP-11 and the M6800 to illustrate machine language and assembly language programming and architectural concepts. It is impossible today to provide examples of programs for all the different computers, especially microcomputers, used in various schools. These two were chosen because they are representative and because they offer especially clean organizations. Using the flowcharts and programming principles given, a student should be able to use this book with any computer. (In any case, the book should be supplemented with a manual or at least a programming card for the computer chosen.) If a laboratory is offered, students may design and construct simple logic circuits, either with a logic trainer or with integrated circuits, in the first half of the course and write and run machine and assembly language programs in the second half of the course.

Additional changes in this second edition include greater emphasis on use of MSI circuits, ROMs, and PLAs in logic design, and discussion of newer computers, including 16-bit microcomputers and the VAX in the chapter on architecture.

The theme of this book is computer structure. The structure of a computer is examined at several progressive levels. Structures at one level become the components of a higher level. We begin at the logic level, skipping over the electronic circuit level, a discussion of which would require the student to have a background in physics, circuit analysis, and electronics that is not needed for the rest of the book. Unlike most other introductory computer hardware texts, this one does not begin with number representation. This topic is instead covered in an appendix, where it may serve as a review for students who are familiar with it; most students today have encountered binary arithmetic in high school or earlier. If desired, an instructor may choose to begin with this appendix. More advanced aspects of number representation are treated in context as they are needed.

The main chapters are prefaced with an overview that allows the student to prepare a mental framework for the material of each chapter in accordance with the principles of cognitive learning theory. Each chapter ends with a summary, references, and problems. The section of each text needed for each problem is shown in parentheses to facilitate assigning problems. Answers for selected problems are given in Appendix B. Since this is an introductory text, most references are to other texts rather than to the original research papers.

Chapter 1 presents an overview of the structural levels of the computer. The logic level discussions (chapters 2 through 5) cover combinational logic circuit analysis and design; logic technologies and integrated circuit implementations of standard logic functions; flipflops and sequential circuits, concentrating on common computer circuits such as registers and counters; and construction of logic systems from register-transfer modules. Chapter 6 consolidates the material on the logic level by discussing the arithmetic circuits needed for a computer. The programming level is treated in chapters 7 through 8, which discuss the machine language and assembly language programming sublevels. Memories and microprogramming, input/output devices and interrupt handling, and the organization of minicomputers and microprocessors are developed in chapters 9, 10, and 11, respectively—the systems level. The back matter includes a review of binary arithmetic, and a glossary.

The discussion in chapters 7 and 8 centers on small computers, both because it is important for students to have hands-on experience with a computer and because such computers are relatively inexpensive and widely available for student use. The discussion of programming is made as general as possible, using the PDP-11 and the M6800 as running examples. Instruction sets of other popular computers are also discussed. The instructor can readily supplement this material with information about any computer available for class use.

Many people helped in the development of this text. Reviewers for this second edition include Edward K. Bowdon, University of Arkansas; C. George Brockus, U.S. Naval Academy; D. R. Clutterham, Florida Institute of Technology; Darrow F. Dawson, University of Missouri-Rolla; Robert N. Dreyer, Chattanooga State Technical Community College; Souhail El-Asfouri, University of Houston; Donald E. Kirk, Naval Post graduate School; Matthew McCann, The University of Steubenville; Gerald Peterson, University of Arizona; Sudhakar M. Reddy, University of Iowa; Gilbert Shaw, University of Oregon; James H. Tracey, University of Colorado, Colorado Springs.

CONTENTS

3 Combinational Logic Circuits and Logic Technologies

4 Sequential Logic

5 Register-Transfer Logic

PART III THE SYSTEMS LEVEL

9 Memory

10 Input/Output

11 Computer Systems

1

Overview

What are you able to build with your blocks?
Castles and palaces, temples and docks.
Robert Louis Stevenson

The theme of this book is the diversity of levels in the structures of digital computers and other digital systems. This chapter defines the levels that we will study and provides an overview of the book.

1.1 INTRODUCTION

Modern digital computers are complex systems. To understand them, we will view them as a hierarchy of system levels, as first described by Bell and Newell (1971). We will consider four basic levels: electronic circuits, logic, programming, and computer systems. (The logic level is further divided into three sublevels: combinational logic, sequential logic, and register-transfer logic.) Since a thorough understanding of electronic circuits requires some advanced knowledge of physics and mathematics, we will concentrate on the other three levels, which are unique to computers and to digital technology.

The levels nest together in a hierarchy. At each level the system may be described in terms of appropriate components, structures, and system behavior. The components at a particular level obey certain physical or mathematical laws and combine to make structures. Together, the components and structures determine the behavior of the system at that level. Each level also has a distinct language or languages to describe components and their connections.

Table 1.1 shows the system levels with examples of their components and structures. Note that structures at one level can become components at another level. For example, gates and flipflops are both structures at the electronic circuit level, but gates are components at the combinational logic sublevel and flipflops are components at the sequential logic sublevel.

The purpose of this book is to study each computer level (except the electronic circuit level) briefly, yet in enough detail to show its characteristics and importance to computer systems. Because digital technology is rapidly changing, not all the levels we will study are well established or well understood. While the lowest levels have been organized or codified for several decades, the newer levels—particularly the register-transfer level and the computer systems level—change constantly. New levels may yet emerge. In addition, the levels described here cannot completely describe computer system behavior, since they exclude mechanical devices, such as card readers, teletype terminals, and line printers. We will study such devices briefly in chapter 10 to see how they interact with other structures and to improve our knowledge of computer hardware.

TABLE 1.1

HIERARCHY OF SYSTEM LEVELS WITH EXAMPLES OF STRUCTURES AND COMPONENTS

LEVEL	STRUCTURES	COMPONENTS
Computer systems	Computers, networks	Controls, processors, memories
Programming	Programs	Instructions, subroutines, memories, operators
Logic		
Register-transfer	Circuits—arithmetic unit	Registers, data operators
Sequential	Circuits—counters, registers	Flipflops—*RS, JK, T, D*
Combinational	Circuits—encoders, decoders	Gates—AND, OR, NOT, NOR, NAND
Electronic Circuit	Circuits—gates, multivibrators (flipflops), amplifiers	Active—transistors, voltage sources Passive—resistors, capacitors, inductors, diodes

The study of computer systems is similar to the study of human systems. Humans, too, can be studied at many levels, for example the anatomical, the neurological, the biochemical, the psychological, the sociological, and the anthropological, to name just a few. An attempt to classify such a study would yield a hierarchy in which some structures at one level, say the biochemical, become components at another level, say the physiological. In other cases, the hierarchy is not clear. Just as there are careers in human systems that involve primarily one level, so there are careers in computer systems that concentrate on one level, such as programming or logic design. Yet a knowledge of the impact of one level on the entire system is important in order to analyze, design, and use systems successfully.

The rest of this chapter contains brief overviews of system levels. All the ideas introduced here will be discussed in more detail in later chapters. Some of the words and concepts may be unfamiliar to you now but will become familiar as you progress through the book. This chapter is intended to provide a framework for understanding the hierarchy of system levels; it is not intended to introduce any level rigorously.

1.2 ELECTRONIC CIRCUIT LEVEL

The lowest level shown in table 1.1 is the electronic circuit level. (A lower level might combine electromagnetic theory and quantum mechanics to explain the operation of the components at this level.) Its components can be passive (such as resistors, capacitors, inductors, and diodes) or active (such as voltage or current sources and transistors). Circuit behavior is characterized by continuous waveforms of voltage and current, which can be described by differential equations.

We will not consider the electronic circuit level in any detail since it requires a knowledge of circuit analysis and physics beyond that assumed for this book. However, we will consider how the technologies chosen to implement electronic circuit design, for example metal-oxide-semiconductor (MOS) technology, affect other levels of computer operation.

1.3 LOGIC LEVEL

The concept of logic level is unique to digital technology. The preceding electronic circuit level, in contrast, is useful for many technologies. The difference between the two levels is most obvious in circuit behavior. At the logic level, circuit behavior is described by discrete binary variables that are called 0 and 1, or low and high, regardless of the exact voltages to which they correspond. At the electronic circuit level, circuit behavior is described by continuous waveforms that require analysis by differential equations, a far more complex form of mathematics.

If operation of a logic circuit depends only on the current values of the inputs, it is called *combinational logic*. The mathematics necessary to describe combinational logic is switching algebra. The components perform logical operations such as logical addition (OR) and logical multiplication (AND), which are analogous to conventional addition and multiplication. (We will define these operations precisely in chapter 2.) The components that implement the operations are connected at their terminals in much the same way that electronic components are connected. The signals are considered to be discrete 0s or 1s.

After learning to analyze and design combinational logic circuits with switching algebra, we will look at some simple map and table methods for combinational logic. We will also look at integrated circuits that have several combinational logic circuits on single chips of silicon. We will consider how integrated circuits have modified older methods of combinational logic design.

The *sequential logic* sublevel characterizes logic circuits whose behavior depends in part on the past history of the circuit. In addition to the logic components used at the combinational logic sublevel, the sequential logic sublevel requires memory or delay components, such as flipflops. A flipflop is a small storage unit whose output can be either 0 or 1. Difference equations, the discrete analog of the differential equations used at the electronic circuit level, describe circuit behavior.

We will study two basic types of sequential logic circuits—*registers* and *counters*. Ordinarily, computers and other digital systems operate with a basic unit of information called a *word*. For a given system a word has a standard number of bits, usually ranging from 4 to 64, where each bit can be either 0 or 1. Registers store information and usually hold one word, but they can be shorter or longer as needed. Counters, as their name implies, count computer operations or time intervals and are basic to the timing and control of a computer.

While both the combinational logic sublevel and the sequential logic sublevel have been standardized for nearly as long a time as electronic computers have existed, the last logic sublevel, the register-transfer level, is new and uncertain. Its components are registers and transfer paths between registers. The behavior of a register-transfer system is represented by the sequence of bit values. Contents of the registers can be combined by logical operations,