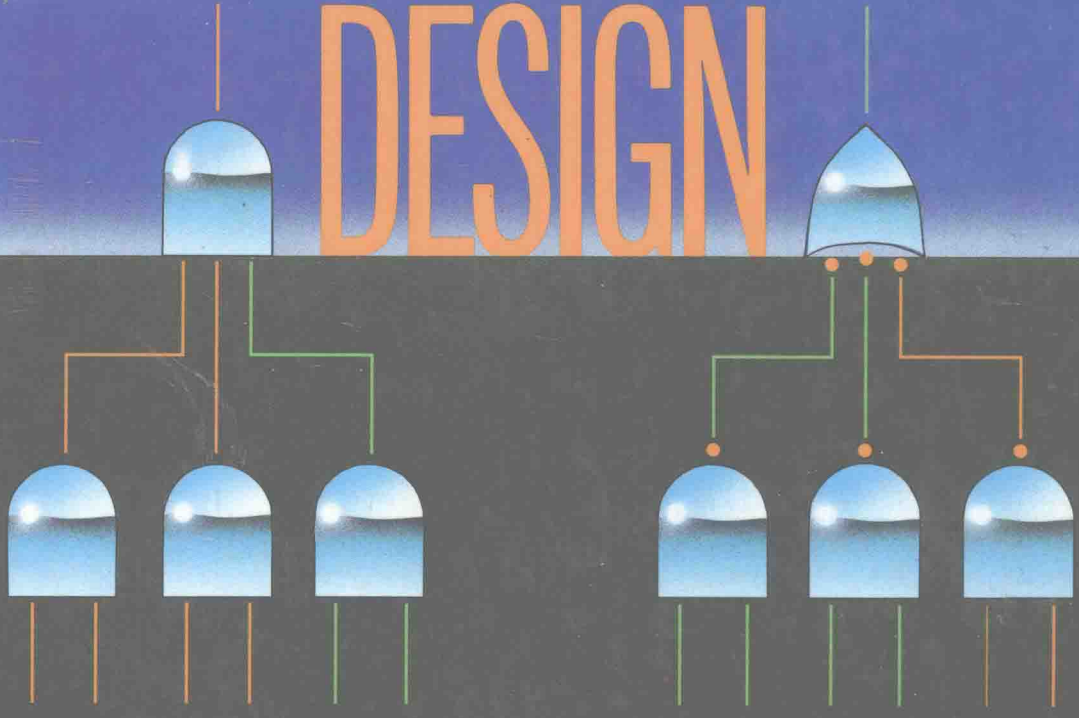
THOMAS C. BARTEE

COMPUTER ARCHITECTURE AND LOGIC DESIGN



COMPUTER ARCHITECTURE AND LOGIC DESIGN

Thomas C. Bartee

Harvard University

McGraw-Hill, Inc.

New York St. Louis San Francisco Auckland Bogotá Caracas
Hamburg Lisbon London Madrid Mexico Milan Montreal
New Delhi Paris San Juan São Paulo Singapore Sydney Tokyo Toronto

To my mother and father

This book was set in Times Roman by Publication Services.
The editor was David M. Shapiro;
the production supervisor was Kathryn Porzio.
The cover was designed by Joseph Gillians.
Project supervision was done by Publication Services.
R. R. Donnelley & Sons Company was printer and binder.

COMPUTER ARCHITECTURE AND LOGIC DESIGN

3 4 5 6 7 8 9 0 DOC DOC 9 5 4 3 2 1

ISBN 0-07-003909-7

Library of Congress Cataloging-in-Publication Data

Bartee, Thomas C.
Computer architecture and logic design / Thomas C. Bartee.
p. cm.
ISBN 0-07-003909-7
1. Computer architecture. 2. Logic design. I. Title.
QA76.9.A73B374 1991
004.2′2—dc20 90-42808

COMPUTER ARCHITECTURE AND LOGIC DESIGN

Also Available from McGraw-Hill

Schaum's Outline Series in Computers

Most outlines include basic theory, definitions, and hundreds of solved problems and supplementary problems with answers.

Titles on the Current List Include:

Advanced Structured Cobol
Boolean Algebra
Computer Graphics
Computer Science
Computers and Business
Computers and Programming
Data Processing
Data Structures
Digital Principles, 2d edition
Discrete Mathematics
Essential Computer Mathematics
Linear Algebra, 2d edition
Mathematical Handbook of Formulas & Tables
Matrix Operations
Microprocessor Fundamentals, 2d edition
Programming with Advanced Structured Cobol
Programming with Assembly Language
Programming with Basic, 3d edition
Programming with C
Programming with Fortran
Programming with Pascal
Programming with Structured Cobol

Schaum's Solved Problems Books

Each title in this series is a complete and expert source of solved problems containing thousands of problems with worked out solutions.

Related Titles on the Current List Include:

3000 Solved Problems in Calculus
2500 Solved Problems in Differential Equations
2000 Solved Problems in Discrete Mathematics
3000 Solved Problems in Linear Algebra
2000 Solved Problems in Numerical Analysis

Available at your College Bookstore. A complete listing of Schaum titles may be obtained by writing to: Schaum Division
McGraw-Hill, Inc.
Princeton Road, S-1
Hightstown, NJ 08520

ABOUT THE AUTHOR

Thomas C. Bartee is Gordon MacKay Professor of Computer Engineering at Harvard University, where he was also Director of the Electronic Design Center. He has also taught at the University of Rhode Island and George Washington University, and his affiliations have included MIT, IDA, IBM, EGG, Raytheon, Hickok, and NUSC. He is past President of the IEEE NE Computer Group, past editor of the IEEE Computer Journal reviews, and an IEEE Distinguished Lecturer. He received an award for outstanding contribution to computer science education from Westminster College. Professor Bartee has published several well-known research papers and books on Computer Design and Communications.

PREFACE

The purpose of this book is to present, as clearly as possible, the principles of modern digital computers. The book covers the major topics, particularly emphasizing logic design and architecture. The latest developments are explained, including programmable logic devices, ASICs, caches, virtual memory, RISCs, and pipelines. Many questions are included at the end of each chapter, and answers to selected questions may be found at the end of the book.

There is considerable information in the book, and it can be used for a course covering one or two semesters or one or two quarters. It has been widely used for one-semester courses on logic design, architecture, organization, and other hardware-related subjects. The book is often used in conjunction with lab work and also with introductory study of microprocessor programming and assembly language.

Chapter 1 presents an introduction to computer usage and modes of operation. A historical background is included, as is some basic information concerning assembly languages, high-level languages, and an overview of system and application programs.

Chapters 2 through 4 introduce the basic ideas and principles used in all digital systems, including instruments, communications systems, and control systems, as well as computers and microprocessors. These chapters explain number systems, logic design, and boolean algebra. The latest devices are used as examples, and custom ICs, ASICs, very-large-scale integration, and programmable logic devices are introduced.

Chapter 5 presents the principles of arithmetic operations in computers. This chapter explains how arithmetic is performed in a computer and how the arithmetic sections of computers are organized.

Chapter 6 introduces modern digital memories, including static and dynamic integrated-circuit memories, tapes, disks, bubbles, CCDs, and optical memories. An introduction to memory organization is included, and the latest material on virtual memories and caches is presented.

Chapters 7 and 8 describe input/ouput devices, such as keyboards, printers, and flat panel displays, and covers the principles of their operation. The organization and operation of input/output circuitry is covered, including buses, I/O processors, and interrupt techniques. I/O design fundamentals are illustrated with examples.

Chapter 9 explains how the operations performed by computers are controlled. The organization of control is covered, as well as hard-wired and microprogrammed control. Register transfer language and other design tools are introduced along with timing charts.

Chapters 10 and 11 describe computer architecture and introduce some representative architectures. This is a broad subject and includes considerable information introduced in earlier chapters. The latest topics in computer architecture are introduced in Chap. 10, including RISCs and CISCs. security, and pipelines. In Chap. 11 information on the development of the major microprocessor lines is presented, and the architectures of several selected microprocessors are discussed.

Chapter 12 introduces the major circuit lines used in modern digital systems. This chapter constitutes a discrete module and can be moved forward in the course of study if desired.

The book adheres to the latest standards, including the ANSI, JEDEC, ISO, Mil, CBEMA, NIST, and IEEE standards for block diagram symbols, floating-point arithmetic, circuit symbols, memory device formats, and IC packages.

All of my teaching and research at MIT, Harvard, the University of Rhode Island, George Washington University, IDA, Raytheon, IBM, EGG, and NUSC has played a part in the development of this book. I will always be indebted to Profs. I.S. Reed, E.J. McCluskey, G. Birkhoff, W.W. Peterson, I.L. Lebow, M.A. Miller, and G. Goff for their assistance and encouragement. The final manuscript was worked on by the following teachers in the United States, Europe, and Asia at the direction of the book's editor, David Shapiro: Lincoln B. Aniteye, Hofstra University; Patty Brayton, University of Oklahoma; Kumar Shiv, University of Missouri–Rolla; Martha Sloan, Michigan Technological University; Paul D. Stigall, University of Missouri–Rolla; and Charles Wright, Iowa State University. Their suggestions are greatly appreciated.

Thomas C. Bartee

CONTENTS

*Sections marked with asterisks can be omitted in a first reading without loss of continuity.

CHAPTER 1

COMPUTER OPERATION

1.1 ELECTRONIC DIGITAL COMPUTERS

The history of attempts to make machines that can perform long sequences of calculations automatically is fairly long. The best-known early attempt was made in the 19th century by Charles Babbage, an English scientist and mathematician. Babbage attempted to mechanize sequences of calculations, eliminating the operator, through construction of a machine that would perform all necessary operations in a predetermined sequence. The machine designed by Babbage used cardboard cards with holes punched in them to introduce both instructions and the necessary data (numbers) into the machine. The machine was to perform the instructions dictated by the cards automatically, not stopping until an entire sequence of instructions had been completed. The punched cards used to control the machine had been used to control the operation of weaving machines. Surprisingly enough, Babbage obtained some money for his project from the English government and started construction. Although he was severely limited by the technology of his time, and the machine was never completed, Babbage succeeded in establishing the basic principles on which modern computers are constructed. There is even some speculation that if he had not run short of money, he might have constructed a successful machine. Although Babbage died without realizing his dream, he established the fundamental concepts used to construct machines elaborate beyond even his expectations.

By the 1930s, punched cards were in wide use in large businesses, and various types of punched-card-handling machines were available. In 1937 Howard Aiken, at Harvard, proposed to IBM that a machine could be constructed (by using some of the parts and techniques from the punched-card machines) that would automatically sequence the operations and calculations performed. This machine used a combination of electromechanical devices, including many relays. The machine was in operation for some time, generating many tables of mathematical functions (particularly Bessel functions), and was used for trajectory calculations in World War II.

Aiken's machine was remarkable for its time but was limited in speed by its use both of relays rather than electronic devices and of punched cards for sequencing the operations. In 1943 S. P. Eckert and J. W. Mauchly, of the Moore School of Engineering of the University of Pennsylvania, started the ENIAC, which used electronic components (primarily vacuum tubes) and therefore was faster, but which also used switches and a wired plugboard to implement the programming of operations. Later Eckert and Mauchly built the EDVAC, which had its program stored in the computer's memory and did not depend on external sequencing. This was an important innovation. A computer that stores its list of operations, or program, internally is called a *stored-program computer*. Actually, the EDSAC, designed by Wilkes at the University of Cambridge, begun later but completed before EDVAC, was the first operational stored-program computer.[1]

A year or so later, John Von Neumann, at the Institute for Advanced Study (IAS) in Princeton, started the IAS computer in conjunction with the Moore School of Engineering, and this machine incorporated most of the general concepts of parallel binary stored-program computers.

The UNIVAC I was the first commercially available electronic digital computer. It was designed by Eckert and Mauchly at their own company, which was later bought by Sperry Rand. The U.S. Board of the Census bought the first UNIVAC. (Later UNIVAC and half of Aiken's machine were placed in the Smithsonian Institution, where they may now be seen.) IBM entered the competition with the IBM 701, a large machine, in 1953, and in 1954 it introduced the IBM 650, a much smaller machine, which was very successful. The IBM 701 was the forerunner of the 704–709–7094 series of IBM machines, the first "big winners" in the large-machine category.

Quite a few vacuum-tube electronic computers were available and in use by the late 1950s, but at this time an important innovation in electronics appeared—the transistor. The replacement of large, expensive (hot) vacuum tubes with small, inexpensive, reliable, comparatively low-heat-dissipating transistors led to what are called *second-generation computers*. The size and importance of the computer industry grew at amazing rates, while the costs of individual computers dropped substantially.

By 1965 a third generation of computers was introduced. (The IBM Corporation, in introducing the 360 series, used the term *third-generation* as a key phrase in their advertising, and it remains a catchword for describing all machines of this era.) The machines of this period began making heavy use of *integrated circuits*, in which many transistors and other components are fabricated and packaged together in a single small container. The low prices and high packing densities of these circuits, combined with lessons learned from prior machines, led to some differences in computer system design. These machines proliferated and expanded the computer industry to its present multibillion-dollar size.

Present-day computers are less easily distinguished from preceding generations. There are some striking and important differences, however. The manufacture of integrated circuits has become so advanced as to incorporate millions of active components

---

[1] The Manchester Mark I appears to have been the first computer to execute a stored program, in 1948, but it was built to list memory devices, not as an operational computer. It was designed by Williams and Kilburn.

in the area of an inch, leading to the levels called *large-scale integration* (LSI) and *very large-scale integration* (VLSI). This has led to small-size, lower-cost, large-memory, ultrafast computers ranging from the familiar personal computers (PCs) to the high-performance, high-priced supercomputers.

1.2 SOME DIFFERENT TYPES OF COMPUTER SYSTEMS

An early use of computers was in operating programs punched into cards or recorded on tape and run by the machine, which would then prepare printouts, checks, or some form of data presentation specifying the results. This is an example of *batch processing*. When a computer is used in this way, the input data (and often the program) are introduced to the computer and processed automatically, generally without operator intervention. Many different jobs (or sets of data) can be processed one right after the other, or even at the same time, without any interaction from the system's user during program operation. For instance, keyboard operators may prepare tapes or disks containing data on customers and claims. A large computer processes the data and issues checks, sends out bills, prints records for the company, and so on.

Similarly, the user of an early computer in a scientific laboratory would enter punched program cards and data cards as a *job*, along with any necessary notes to the operators of the machine. A number of these decks of cards were collected, stacked, and finally run by the computer. Later the results were printed and, perhaps even the next day, delivered to the individual users. This also constitutes batch processing.

In other types of systems, users interact with the computer directly, inserting and receiving the data as desired. For instance, an airline ticket agent wishing to make a reservation types the desired flight number and passenger identification on a special typewriter, which communicates via the telephone lines with a computer. The computer looks in its memory, sees whether the flight is full, and, if not, enters the passenger's name on its list for the flight; it then communicates this fact back to the airline ticket agent. If no seats are available, the computer sends this information to the ticket agent, who attempts to interest the passenger in another flight. In this way, the computer connects all ticket agents, keeping a constant record of flights, passengers, and payments and doing all the bookkeeping. The terminals where the ticket agents are located are scattered throughout the world but communicate with the computer via telephone lines. Motels, hotels, stockbrokers, and many businesses have similar systems for reservations, information transferral, and bookkeeping.

These are called *interactive* systems, for the users of the system communicate directly with the computer and the computer responds directly. The development of these systems has progressed in parallel with the development of keyboard input devices as well as output devices of various types, including video monitors which use cathode-ray tubes (CRTs) and printers.

Interactive systems are widely employed in scientific applications, where users operate their programs at a terminal connected to the computer, perhaps by telephone lines, making changes and variations at will. Experimenters can try a set of inputs and study the results, then try other inputs and study those results. The technique of interactive computing is used by circuit designers, architects, chemists, and almost any other scientific area, including medical systems for hospital use.

Personal, or home, computers are also examples of interactive systems, since the user communicates directly with the computer, inputting data and sometimes programs and directly receiving results. Figure 1.1 shows a personal computer with a keyboard and video display.

A much-used input/output device is the *terminal*. A terminal consists of a keyboard and either a CRT display, which is often called a video display unit (VDU), or a printer. The CRT-keyboard combination is often called a *video terminal*, or a *VDT* (for video display terminal). The printer-keyboard combination is referred to as a *hard-copy terminal* or *printer terminal*.

Terminals can be *dumb*, which means that they cannot compute and have no meaningful memory, or *smart*, in which case they include a microcomputer and are in effect a PC. A terminal-PC combination with a powerful graphics capability is often called a *workstation*.

In some systems, the console terminal is located some distance from the computer. A special attachment called a *modem* is used, which makes it possible to transmit the electrical signals generated by the terminal to the computer and to receive the computer's response over telephone lines. At the computer is another modem, which also can transmit or receive. This allows communication in both directions over telephone lines. The user of the terminal simply dials the number for the computer, establishes a connection, provides his or her identity and authorization to use the computer (the computer generally checks a password), and then proceeds to use the computer.

Terminal characteristics are fairly standardized; the same terminal can commonly use any of several different computers. Many companies now provide computer services to users who have terminals or PCs at their disposal. The users simply dial the computer they prefer.

When a number of users share a computer, sometimes via telephone lines, at the same time, the computer is said to be *timeshared*. This means that the computer is able to alternate and interweave the running of its programs so that several jobs or users

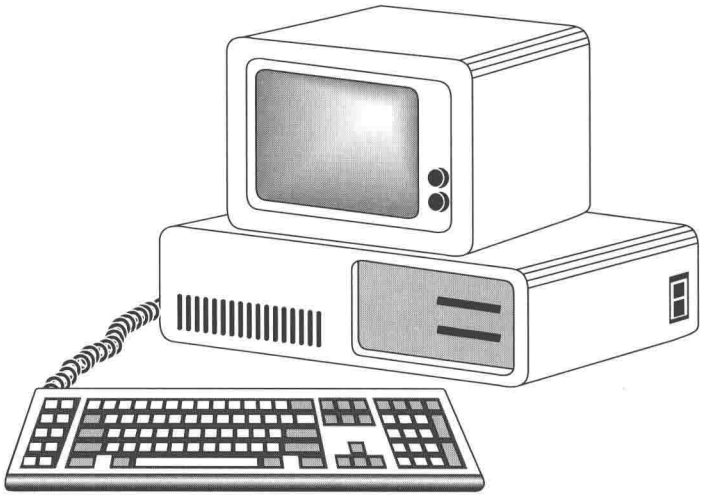


FIGURE 1.1
A personal computer with keyboard and video display unit (VDU).