

---

# LOGIC AND COMPUTER DESIGN FUNDAMENTALS

---

M. Morris Mano

California State University, Los Angeles

Charles R. Kime

University of Wisconsin, Madison



Prentice Hall  
Upper Saddle River, New Jersey 07458

**Library of Congress Cataloging-in-Publication Data**

Mano, M. Morris,

Logic and computer design fundamentals / M. Morris Mano and  
Charles R. Kime.

p. cm.

Includes bibliographical references and index.

ISBN 0-13-182098-2

1. Electronic digital computers--Circuits. 2. Logic circuits.

3. Logic design. I. Kime, Charles R. II. Title.

TK7888.4.M36 1997

621.39'2--dc20

96-26502

CIP

Editor-in-Chief: **Marcia Horton**

Acquisitions editor: **Tom Robbins**

Managing editor: **Bayani Mendoza de Leon**

Production editor: **Irwin Zucker**

Art director: **Amy Rosen**

Assistant art director: **Rod Hernandez**

Creative director: **Paula Maylahn**

Cover design: **Bruce Kenselaar**

Cover illustration: **David Bishop**

Interior design: **Judy Matz-Coniglio**

Manufacturing buyer: **Donna Sullivan**

Editorial assistant: **Nancy Garcia**



© 1997 by Prentice-Hall, Inc.

Simon & Schuster / A Viacom Company

Upper Saddle River, New Jersey 07458

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.

**Trademark Information**

Actel and ACT are registered trademarks of Actel Corporation. Altera and MAX are  
registered trademarks of Altera, Inc. MAX 7000 is a trademark of Altera  
Corporation. PAL is a registered trademark of Advanced Micro Devices, Inc. Viewlogic,  
ViewDraw and ViewSim are registered trademarks of Viewlogic Systems, Inc. ViewTrace is a  
trademark of Viewlogic Systems, Inc. Xilinx is a registered trademark of Xilinx, Inc.  
XC4000 is a trademark of Xilinx, Inc.

The author and publisher of this book have used their best efforts in preparing this book.  
These efforts include the development, research, and testing of the theories and programs to  
determine their effectiveness. The author and publisher make no warranty of any kind,  
expressed or implied, with regard to these programs or the documentation contained in this  
book. The author and publisher shall not be liable in any event for incidental or  
consequential damages in connection with, or arising out of, the furnishing, performance, or  
use of these programs.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-182098-2

Prentice-Hall International (UK) Limited, London

Prentice-Hall of Australia Pty. Limited, Sydney

Prentice-Hall Canada Inc., Toronto

Prentice-Hall Hispanoamericana, S.A., Mexico

Prentice-Hall of India Private Limited, New Delhi

Prentice-Hall of Japan, Tokyo

Simon & Schuster Asia Pte. Ltd., Singapore

Editora Prentice-Hall do Brasil, Ltda., Rio de Janeiro

# CONTENTS

## PREFACE xv

## □ CHAPTER 1

### DIGITAL COMPUTERS AND INFORMATION 2

- 1-1 **DIGITAL COMPUTERS 3**
  - Information Representation 5
  - Computer Structure 6
  - More on the Generic Computer 6
- 1-2 **NUMBER SYSTEMS 8**
  - Binary Numbers 9
  - Octal and Hexadecimal Numbers 11
  - Number Ranges 12
- 1-3 **ARITHMETIC OPERATIONS 13**
  - Conversion from Decimal to Other Bases 15
- 1-4 **DECIMAL CODES 17**
  - BCD Addition 19
- 1-5 **ALPHANUMERIC CODES 20**
  - ASCII Character Code 20
  - Parity Bit 22
  - Unicode 23
- 1-6 **CHAPTER SUMMARY 26**
  - REFERENCES 26**
  - PROBLEMS 27**

## □ CHAPTER 2

### COMBINATIONAL LOGIC CIRCUITS 30

- 2-1 **BINARY LOGIC AND GATES 31**
  - Binary Logic 32
  - Logic Gates 33

## CHAPTER

# 1

# DIGITAL COMPUTERS AND INFORMATION

Logic design fundamentals and computer design fundamentals are the topics of this book. Logic design deals with the basic concepts and tools used to design digital hardware consisting of logic circuits. Computer design deals with the additional concepts and tools used to design computers and other complex digital hardware. Computers and digital hardware in general are referred to as digital systems. Thus, this book is about understanding and designing digital systems. Due to its generality and complexity, the computer provides an ideal vehicle for learning the concepts of and tools for digital system design. In addition, due to its widespread use, the computer itself is deserving of study. Hence, the focus in this book is on computers and their design.

The computer will be not only a vehicle, but also a motivator for study. To this end, we use the exploded pictorial diagram of a computer of the class commonly referred to as a PC (personal computer) given on the opposite page. We use this generic computer to highlight the significance of the material covered and its relationship to the overall system. A bit later in the chapter, we will discuss the various major components of the generic computer and see how they relate to a block diagram often used to describe a computer.

### 1-1 DIGITAL COMPUTERS

Today, digital computers have such a prominent and growing role in modern society, that we often say we are in the “information age.” Computers are involved in our business transactions, communications, transportation, medical treatment, and entertainment. They monitor our weather and environment. In the industrial world,

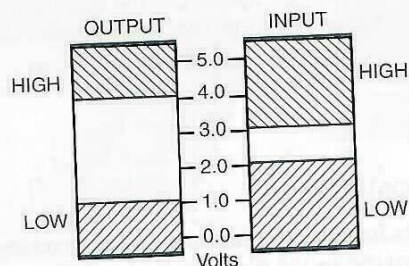
they are heavily employed in design, manufacturing, distribution, and sales. They have contributed to many scientific discoveries and engineering developments that would have been unattainable otherwise. Notably, the design of a new processor for a modern computer could not be done without the use of many computers!

The most striking property of the digital computer is its generality. It can follow a sequence of instructions, called a program, that operates on given data. The user can specify and change the program or the data according to specific needs. As a result of this flexibility, general-purpose digital computers can perform a variety of information-processing tasks that range over a very wide spectrum of applications. The general-purpose digital computer is the best known example of a *digital system*. Characteristic of a digital system is its manipulation of discrete elements of information. Any set that is restricted to a finite number of elements contains discrete information. Examples of discrete sets are the 10 decimal digits, the 26 letters of the alphabet, the 52 playing cards, and the 64 squares of a chessboard. Early digital computers were used mostly for numeric computations. In this case, the discrete elements used were the digits. From such an application, the term *digital computer* emerged.

Discrete elements of information are represented in a digital system by physical quantities called *signals*. Electrical signals such as voltages and currents are most common. Electronic devices called transistors predominate in the circuitry that implements these signals. The signals in all present-day electronic digital systems use just two discrete values and are therefore said to be *binary*.

We typically represent the two discrete values by ranges of voltage values called HIGH and LOW. Output voltage ranges and input voltage ranges are illustrated in Figure 1-1. The HIGH output voltage value ranges between 4.0 and 5.5 volts, and the LOW output voltage value ranges between -0.5 and 1.0 volt. The high input range allows 3.0 to 5.5 volts to be recognized as a HIGH, and the low input range allows -0.5 to 2.0 volts to be recognized as a LOW. The fact that the input ranges are longer than the output ranges allows the circuits to function correctly in spite of variations in their behavior and undesirable "noise" voltages that may be added to or subtracted from the outputs.

We give the output and input voltage ranges a number of different names. Among these are HIGH and LOW, TRUE and FALSE, and 1 and 0. It is clear that



□ **FIGURE 1-1**  
An Example of Voltage Ranges for Binary Signals

the higher voltage ranges are associated with HIGH, abbreviated H, and the lower voltage ranges with LOW, abbreviated L. We find, however, that for TRUE and 1 and FALSE and 0, there is a choice. TRUE and 1 can be associated with either the higher or lower voltage range and FALSE and 0 with the other range. Unless otherwise indicated, we assume that TRUE and 1 are associated with the higher of the voltage ranges, H, and that FALSE and 0 are associated with the lower of the voltage ranges, L.

Why is binary used? In contrast to the situation in Figure 1-1, consider a system with 10 values representing the decimal digits. In such a system, the voltages available—say, 0 to 5.0 volts—could be divided into 10 ranges, each of length 0.5 volt. A circuit would provide an output voltage within each of these 10 ranges. An input of a circuit would need to determine in which of the 10 ranges an applied voltage lies. If we wish to allow for noise on the voltages, then output voltage might be permitted to range over less than 0.25 volt, and boundaries between inputs could only vary by less than 0.25 volt. This would require complex and costly electronic circuits and still could be disturbed by small “noise” voltages or small variations in the circuits occurring during their manufacture or use. As a consequence, the use of such multivalued circuits is very limited. Instead, binary circuits are used in which correct circuit operation can be achieved with significant variations in both the two output voltages and the two input ranges. The resulting transistor circuit with an output that is either HIGH or LOW is simple, easy to design, and extremely reliable.

### Information Representation

Since 0 and 1 are associated with the binary number system, they are the preferred names for the signal ranges. A binary digit is called a *bit*. Information is represented in digital computers by groups of bits. By using various coding techniques, groups of bits can be made to represent not only binary numbers, but also other groups of discrete symbols. Groups of bits, properly arranged, can even specify to the computer the instructions to be executed and the data to be processed.

Discrete quantities of information either emerge from the nature of the data being processed or may be purposely quantized from continuous values. For example, a payroll schedule is inherently discrete data containing employee names, social security numbers, weekly salaries, income taxes, and so on. An employee's paycheck is processed using discrete data values such as letters of the alphabet (for the employee's name), digits (for the salary), and special symbols like \$. On the other hand, an engineer may measure the speed of rotation of an automobile wheel, which varies continuously with time, but may record only specific values at specific times in tabular form. The engineer is thus quantizing the continuous data, making each number in the table a discrete quantity of information. In a case such as this, if the measurement can be converted to an electronic signal, the quantization of the signal in both value and time can be performed automatically by an analog-to-digital conversion device.

## Computer Structure

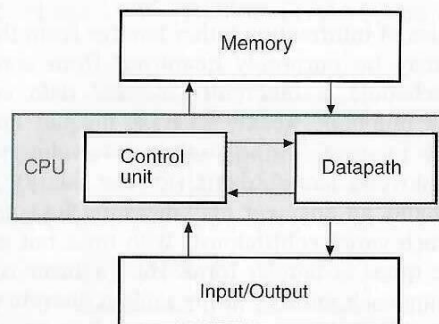
A block diagram of a digital computer is shown in Figure 1-2. The memory stores programs as well as input, output, and intermediate data. The datapath performs arithmetic and other data-processing operations as specified by the program. The control unit supervises the flow of information between the various units. A datapath, when combined with the control unit, forms a component referred to as a *central processing unit*, or CPU.

The program and data prepared by the user are transferred into memory by means of an input device such as a keyboard. An output device, such as a CRT (cathode-ray tube) monitor, displays the results of the computations and presents them to the user. A digital computer can accommodate many different input and output devices, such as hard disks, floppy disk drives, CD-ROM drives, and scanners. These devices use some digital logic, but often include analog electronic circuits, optical sensors, CRTs or LCDs (liquid crystal displays), and electromechanical components.

The control unit in the CPU retrieves the instructions, one by one, from the program stored in the memory. For each instruction, the control unit manipulates the datapath to execute the operation specified by the instruction. Both program and data are stored in memory. A digital computer is a powerful system. It can perform arithmetic computations, manipulate strings of alphabetic characters, and be programmed to make decisions based on internal and external conditions.

## More on the Generic Computer

At this point, we will briefly discuss the generic computer and relate its various parts to the block diagram in Figure 1-2. At the lower left of the picture on page 2 is the heart of the computer, an integrated circuit called the *processor*. Modern processors such as this one are quite complex and consist of millions of transistors. The processor contains four functional modules: the CPU, the FPU, the MMU, and the internal cache.



□ **FIGURE 1-2**  
Block Diagram of a Digital Computer

We have already discussed the CPU. The FPU (floating-point unit) is somewhat like the CPU, except that its datapath and control unit are specifically designed to perform floating-point operations. In essence, these operations process information represented in the form of scientific notation, e.g.,  $1.234 \times 10^7$ , permitting the generic computer to handle very large and very small numbers. The CPU and the FPU, in relation to Figure 1-2, each contain a datapath and a control unit.

The MMU is the memory management unit. The MMU plus the internal cache and the separate blocks near the bottom of the computer labeled "External Cache" and "RAM" (random access memory) are all part of the memory in Figure 1-2. The two caches are special kinds of memory that allow the CPU and FPU to get at the data to be processed much faster than with RAM alone. RAM is what is most commonly referred to as memory. As its main function, the MMU causes the memory that appears to be available to the CPU to be much, much larger than the actual size of the RAM. This is accomplished by data transfers between the RAM and the hard disk shown at the top of the picture of the generic computer. So the hard disk, which we discuss later as an input/output device, appears conceptually as a part of the memory and input/output.

The connection paths shown between the processor, memory, and external cache are the pathways between integrated circuits. These are typically implemented as fine copper conductors on a printed circuit board. The connection paths below the bus interface are referred to as the processor bus. The connections above the bus interface are referred to as the input/output (I/O) bus. The processor bus and the I/O bus attached to the bus interface carry different numbers of bits of data and have different ways of controlling the movement of data. They may also operate at different speeds. The bus interface hardware handles these differences so that data can be communicated between the two buses.

All of the remaining structures in the generic computer are considered part of I/O in Figure 1-2. In terms of sheer physical volume, these structures dominate. In order to enter information into the computer, a keyboard is provided. In order to view output in the form of text or graphics, a graphics adapter card and CRT monitor are provided. The hard disk discussed previously is an electromechanical magnetic storage device. It stores large quantities of information in the form of magnetic flux on spinning disks coated with magnetic materials. In order to control the hard disk and transfer information to and from it, a disk controller is used. The keyboard, graphics adapter card, and disk controller card are all attached to the I/O bus. This allows these devices to communicate through the bus interface with the CPU and other circuitry connected to the processor buses.

The generic computer consists mainly of an interconnection of digital modules. To understand the operation of each module, it is necessary to have a basic knowledge of digital systems and their general behavior. Chapters 1 through 5 of this book deal with logic design of digital circuits in general. Chapters 6 through 8 discuss the primary components of a digital system, their operation, and their design. The operational characteristics of RAM are explained in Chapter 6. The organization and design of datapaths are discussed in Chapter 7. Methods for designing control units are introduced in Chapter 8. Chapters 9 through 11 present

the basics of computer design. Some typical instructions employed in digital computer CPUs are presented in Chapter 9. The organization and design of typical CPUs are examined in Chapter 10. Input and output devices and the various ways that a CPU can communicate with them are discussed in Chapter 11. Finally, memory hierarchy concepts related to the caches and MMU are introduced in Chapter 12.

To guide the reader through this material and to keep in mind the “forest” as we carefully examine many of the “trees,” at the beginning of each chapter we provide a copy of the generic computer. Appropriate blue shading represents the coverage of the material in the chapter. In addition, accompanying discussion appears in a blue box to tie the topics in the chapter to the associated components of the computer. At the completion of our journey, we will have covered most of the various modules of the computer and will have an understanding of the fundamentals that underlie both its function and design.

Earlier we mentioned that a digital computer manipulates discrete elements of information and that all information in the computer is represented in binary form. Operands used for calculations may be expressed in the binary number system or in the decimal system by means of a binary code. The letters of the alphabet are also converted into a binary code. The purpose of the remainder of this chapter is to introduce the binary number system, binary arithmetic, and selected binary codes as a basis for further study in the succeeding chapters. In relation to the generic computer, this material is very important and spans all of the components except some in I/O that involve mechanical operations and analog (as contrasted with digital) electronics. Thus, almost all of the generic computer is shaded in blue.

## 1-2 NUMBER SYSTEMS

The decimal number system is employed in everyday arithmetic to represent numbers by strings of digits. Depending on its position in the string, each digit has an associated value of an integer raised to the power of 10. For example, the decimal number 724.5 is interpreted to represent 7 hundreds plus 2 tens plus 4 units plus 5 tenths. The hundreds, tens, units, and tenths are powers of 10 implied by the position of the digits. The value of the number is computed as follows:

$$724.5 = 7 \times 10^2 + 2 \times 10^1 + 4 \times 10^0 + 5 \times 10^{-1}$$

The convention is to write only the digits and infer the corresponding powers of 10 from their positions. In general, a decimal number with a decimal point is represented by a string of coefficients:

$$A_n A_{n-1} \dots A_1 A_0 . A_{-1} A_{-2} \dots A_{-m+1} A_{-m}$$

Each coefficient  $A_i$  is one of 10 digits (0, 1, 2, 3, 4, 5, 6, 7, 8, 9). The subscript value  $i$  gives the position of the coefficient and, hence, the weight  $10^i$  by which the coefficient must be multiplied.