PRENTICE-HALL EE SERIES

Flores

Computer Logic

The Functional Design of Digital Computers

IVAN FLORES

Consultant
Norwalk, Connecticut

COMPUTER LOGIC

The Functional Design of Digital Computers

Prentice-Hall, Inc.

Englewood Cliffs, New Jersey

1960

PREFACE

This book considers computers first at the overall or highest level of organization and proceeds later to the lower levels of detail. We emphasize the organization and functional interrelationship of the fundamental units. These relationships are discussed from the viewpoint of necessity—they are not dependent upon mathematical logic or Boolean algebra, and they are not arrived at by circuit theory. They are, instead, discussed from the point of view of operational necessity.

A minimum is required of the reader in the way of engineering and mathematical background. The work should be comprehensible to those of some scientific training. Problems have been included to provide a means of acquiring facility with computer theory.

It is the method of the first part of the book to examine computers from the air down—a first look, a bird's-eye view from the air, is taken to see how the computer fits into the overall system of scientific investigation and business enterprise. The characteristics and specifications, then the relationship between the problem and the computer are examined. The structure and organization are investigated with regard to the large functional units, their interrelation, interdependence, and control, and an examination of programming is included. The function and composition of each unit are then pursued. The analytic approach in the first part of the book will help the reader to understand at each step the overall pattern. The second half continues, by synthesis, to build up larger and larger functional units with the goal of a complete and unified machine concept.

The interplay between design and application is emphasized. A design which keeps the user, the programmer, and the applications in mind cannot help but be superior. Similarly, the programmer who is aware of how the computer operates to carry out his instructions is able to make use of the machine.

How is each process initiated and controlled and what are the methods of coordinating the varied activities of the units which comprise the computer?

Programming is considered from the point of view of the programmer, the functional units of the computer, and the type of problem to be handled by the computer and the designer. The job of the programmer and the art of doing it are explained by programming a fictitious machine. The flow diagram is used as a tool. The techniques of tallying, comparison, subroutines, branch points, cycle indices, sentinels, iterations, decision making and automatic program alteration are the programming background required for effective computer design.

The computer performs arithmetic by addition, complementation, and shifting. How these are used to do subtraction, multiplication, and division is explained with examples in the decimal system so that the reader may follow with ease.

A thorough background of counting and number systems is developed in order to bring an understanding of how arithmetic is performed by the computer in its own language. This understanding is expanded to a perception of arithmetic and translation within and between different number systems. The other languages used by machines, the machine codes, and the means for performing arithmetic in machine code are discussed.

Manipulation of five kinds of elements is used to present the process of symbolic logic: propositions, geometric areas, letter symbols, functional blocks, and components. Symbols proposed by the Institute of Radio Engineers' Standards Committee are adopted. After an introductory treatment, logical methods are discussed and the binary half-adder and adder are derived; examples of its use are given. Boolean algebra is summarized. *Nor* circuits are given attention as logical blocks, along with examples of their incorporation in a computer logic.

The logical construction of functional units and operational units from elementary logical units is considered, including a discussion of serial, serial parallel, and parallel adders for different codes and counting systems, accumulators, shift registers, comparators, complementors, coding units, and control units. This material comprises a fund of "unwritten art" in the computer field.

Here is where the *synthesis* approach really pays off. Starting from simple logical elements, large functional units are built up; a simple symbolic block is used to represent each large functional unit. When constructing still larger operational units it is thus possible to see and understand the relationships among the functional units because each is represented by a single block. There is not a welter of symbols all over the page to confuse the reader and obscure and complicate the reasoning.

The section on arithmetic capitalizes upon this principle. The functional and logical blocks are combined before the reader's eyes to make operational units to do each of the arithmetic operations. Before each construction, the reasoning and the elementary steps required of each unit are explained both verbally and with subcommand operational flow charts. The generation of the sign of the result is also covered.

Nothing could be more important to the computer than how its functions are controlled! A chapter is devoted to the previously neglected topic of control unit operation—where the asynchronous decentralized computer control is dissected for the reader. The plan of action for the control unit is presented verbally and with operational flow charts; then each subdivision of control is analyzed. Finally, the logic to supplement the control is presented and discussed; centralized control is also examined here.

The operation of the memory is dependent largely upon the characteristics of the components comprising it. These components are investigated first and the logic necessary for memorizing and remembering follows. More attention is given core and drum memories than others because of their importance in modern computers.

Input and output equipment have received the greatest slight in the computer literature, at least in the aspect of the logic associated with them. Maybe the author has gone overboard in his coverage of this area. But there has been so little available elsewhere that an extensive treatment is necessary to tie things together. Much mechanical detail has been included which may seem expendable. Yet, to appreciate the symbiosis of the mechanical and electronic aspects of the equipment one must comprehend its mechanical function as well as the logical principles involved. The kinds of direct communication to and from the computer are catalogued and control panels of current computers are used to demonstrate these general principles.

To encapsulate the ideas which pervade the book, a problem is presented in the last chapter. A complete analysis is made of what happens from the time the problem is given to the computer programmer until he returns the answer to whomever posed it.

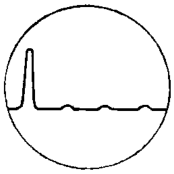
The Glossary which follows the text has been culled from a number of glossaries published by the various professional computer societies. To these extracts have been added a number of definitions not found elsewhere, and some of the published definitions.

I would like to express sincere gratitude to those who have made this venture possible. First, to my wife Helen, who courageously and devotedly waded far afield into the morass of computer technology. Next, to the small band of associates, Saul Teichman, Andre Godefroy, Tom Cull, and Ralph Townsend, who painstakingly read each draft to point out where the text became unintelligible and obscure. Then to the alert and untiring efforts of the stenographic force, notably Virginia Kirchhoff and also Alice Bennett and Evelyn Davison. Lastly, to all my friends who have gracefully accepted my hibernation for the duration of this project.

I. F.

CONTENTS

ONE



INTRODUCTION

1.1. WHAT IS A COMPUTER?

Anyone who has sufficient interest in computers to search out a book on that topic must have some idea of what one is. But somehow a fresh look at this question before plunging into a detailed study may be an advantage.

A computer, whether analog or digital, is a device for solving problems automatically. The word "automatically" indicates that there is as little human intervention as possible. Modern computers can solve problems so quickly because they are electronic devices. The types of problems computers can solve are discussed in the next section.

An important consideration is the language used by the computer. It is obvious that the computer doesn't use the same language as we do, either for receiving the problem or for solving it. If it did, we could merely tell our problem to the computer and it would answer us in spoken English.

The computer uses a unique language for solving problems. A different language may be required to get information into the computer. Thus, we may have to contend with three different languages: the English language we speak; the intermediate language used to convey information to the computer, such as the language of the holes on punched cards; and the language the computer uses for the information and for solving the problem. Let's hope this language barrier is not too great for the reader!

The computer can perform certain *operations* upon the information furnished it. If these operations are repeated in various sequences, different *problems* can be solved. Each *application* requires the solution of one or more problems. Thus a hierarchy exists: a few operations can solve many problems, and sets of these problems make up the applications.

In the next section the problems are discussed in answer to the question, "What can the computer do?" The operations the computer performs are discussed in Section 1.3. Section 1.4 mentions a number of applications.

1.2. WHAT CAN THE COMPUTER DO?

Computers in general can do a number of different things. The list we are going to make may lie beyond the capabilities of any one computer, but there are single computers that can do several of the things listed below. The more kinds of jobs we wish to have the computer do, the less efficiently it will be able to do any single job. On the other hand, if we only wish the computer to be able to do a few jobs, it may be able to do each in less time. In other words, more diverse requirements for the machine result in its taking a longer time to solve a given problem. Examples of tasks that various computers can do follow.

Mathematical Problems

Solutions can be found for linear equations, linear differential equations, matrices, partial differential equations, polynomials, and so forth.

Simulation

An important function of a computer is to act as, or to create, a model of an experimental or practical situation. This can be done in two fashions:

REAL-TIME SIMULATION. Here, whatever should happen in a laboratory or in the field must appear to happen at the *same rate* within the computer.

SCALED SIMULATION. Here the rate of time of simulation within the computer may be foreshortened, so that what would normally take a week or a month to transpire in the field would happen within a few moments in the computer. Or, conversely, the computer may perform in the period of an hour what happens within a split second in actual practice.

Both these forms of simulation are very important, and great pains may be required to get the proper scaling (ratio of computer time to actual time). An example of real-time simulation may help the reader visualize what is meant.

Take the case of a nuclear reactor. Of course, the computer does not simulate the physical characteristics of the reactor—it does not look like it, nor does it in any physical sense act like the reactor. What it does do is produce numbers which represent the energy being generated by such a reactor. The operator of the computer can then note at any moment the state of the reactor and his own capacity to control it as limited by his reaction time. He can find out how the reactor would operate, for instance, if the control rod were fully withdrawn and the reactor went critical. In the field the reactor might blow up; the computer responds by producing a very large number. If the operator at any time wished to examine closely what was happening in a short period of time, he would use scaled simulation instead of real-time simulation.

Control

One of the great accomplishments of computer science is its enabling computers to control things. Not only can a computer control processes and movement outside itself, but it can also control itself!

Some of the areas of control follow.

PROCESSES. Information is supplied to a computer about the functioning of, say, a chemical plant by describing in quantitative form the pressure, temperature, rate of flow, and other data. The computer can then, by the various formulas incorporated within it, determine what adjustments should be made in each of these variables so that the processing is carried on under the proper environmental conditions. It will send back information in the form of currents and voltages or mechanical movements in order to provide compensations in such things as actuators, valves, relays, heating elements, and so forth, and thus effect a control of the processes. The use of such devices and computations provides precise chemical mixtures to produce the delicate fibers required for some of the inexpensive but beautiful fabrics manufactured today.

MOVEMENT. By the same means—by reporting to the computer information on location and rate of movement—we can have the computer produce outputs that control movement. An example of such control can be found in the new "electronic elevators" which have computers that operate to schedule them during periods of even the heaviest traffic in the world's largest skyscrapers. Also, the mechanism that enables the much-talked-about guided missiles to reach their targets is a calculator which can compensate for the various physical phenomena which impinge upon the missile as it flies its path.

ITSELF. The ability of the computer to control itself enables it to follow a given course of action until it has found a particular kind of result and then to adjust its behavior accordingly. The ability to make decisions is what makes the more advanced computer particularly effective.

Record Keeping

One of the sections of a computer which we will discuss in more detail later is its memory or storage facility. Some computers can store large amounts of information in a small physical volume and in a form quickly accessible to the computer. The computer can insert and withdraw information from storage within an extremely short time. This enables the computer to be used for tasks such as keeping track of large inventories for commercial installations or of the entire accounts receivable or accounts payable of a vast organization.

Construction of Tables

Modern computers can perform many arithmetic operations in extremely short periods of time. This computational speed makes it practical and, nowadays, customary to use a computer for the calculation of mathematical and physical tables. The computer exercises control over itself by continuing to change the various look-up variables it uses in calculating the entries for a table.

Language Translation

Because of its ability to store large amounts of information, a big computer can be and is used to translate from one language to another. In order that this translation be more than literal, various complex rules of syntax and grammar must also be stored in the memory of the computer. At the present time it is possible to translate foreign languages by computer for less than it would cost for a human to translate them. But this is so only when the input cost is not included. That's the catch! It is almost as expensive to have an operator enter the material into the machine as it is to have a person translate it!

The high-speed computer is being used now on the immense job of compiling and indexing each and every word in the recently unearthed Dead Sea Scrolls. Many of the scrolls are merely fragments. With this index containing every context in which a word appears, it is possible to do the heretofore unapproachable job of reconstructing these fragments into cogent texts; this work is now in progress.

Function Plotting

The output of a computer can be displayed in graphic form. Thus, in addition to the normal output of the computer which might take the form of printed records or tables or the control of physical phenomena, output can be obtained on a visual basis.

Synthesis and Analysis

An important form of drudgery taken over by the computer calls upon its ability to find the functional relation that exists within raw data. The computer can find the curve which most closely fits given data as long as it is provided with some criterion (such as "least squares") for evaluation. It can go even further than that—it can optimize variables existing in a given problem. This is done for many power companies. A power company has several generators, power lines, and busses; at any time during the day there are different loads on the different lines connected to the generators. Several different makes of computers have been put into use for the power companies to calculate which generators should be running and at what per cent of their maximum rate.

Limitations

It might appear that the computer is a modern magic genie. Press the button and the imp ferrets out your problem and solves it. Not so! There is much hard work involved in systematizing and translating the problem into the computer's limited language and in "telling" the computer just what and when to do things. This will be clarified in the next two chapters.

1.3. HOW DOES IT WORK?

What the computer does to the information it receives is called *processing*. Processing is defined by enumerating the operations which are so classified.

One function of the computer is to *store information.* Computers have provision for temporary storage as well as for long-term storage.

The computer is capable of *altering the information.* This is done in two ways, known as *editing* and *arithmetic* (discussed further below).

The computer is able to *move information about.* Movement takes place from the input device to the computer, from the computer to the output device, and within the computer to the various sections of the computer where storage, editing, and arithmetic take place.

Arithmetic—What Is It?

Because we are all so familiar with how arithmetic is done, it may be difficult to put into precise terms what we require of a computer when we wish it to do arithmetic. For that reason space is devoted here to an explicit statement of what is meant by arithmetic in general.

Arithmetic can best be described in terms of symbols. Symbolically, arithmetic may be represented as "$a \otimes b = c$" where $\otimes$ represents one of the four arithmetic operations (addition, subtraction, multiplication, or division) and a, b, and c are information elements acceptable to the computer. The computer can deal with certain elements only. These restrictions will be discussed subsequently.

Arithmetic can be considered as a mapping relationship. For any two acceptable elements, a and b, and a process, $\otimes$, a third element is determined, c. The relationship is referred to as mapping because one dimension (c) is used to represent three dimensions (a, b, $\otimes$).

We are more familiar with maps representing a three-dimensional locality in two dimensions. On such a map a building is represented by a dot or square. This symbol represents all the floors and ceilings and the roof and basement of the building; it also represents the pipes and the sewers below it and maybe even the subway station below them. Thus, all these things are "mapped" into one symbol. This is called a many-to-one mapping or relationship. Each of the many items (floors, ceilings, and so on) we map into one symbol (the dot or square); and conversely, the symbol stands for many different height levels.

Arithmetic is a many-to-one relationship. Thus the elements 3 and 5 and the process of addition yield the resulting element 8; the elements 4 and 4 and the process of addition yield the element 8; the elements 2 and 4 and the process of multiplication yield the element 8. There are many different possibilities or combinations of choice for the elements a and b and the process $\otimes$ that would yield the same result, 8. It may be said that these various combinations are mapped into the element 8.

Many-to-one relationships exist when there are many combinations of elements and processes that have but one resulting element associated with them, and, conversely, when one result is associated with many different combinations of elements and processes.

Arithmetic—How Is It Done?

The mapping view may be simplified a little if we fix a value in the process dimension and say that for a given process, any two elements may be mapped into a third element. Thus two dimensions (the numbers) are mapped into one dimension. This mapping can be performed by the use of a table. To compute by this method would require four tables, one for each of the processes of arithmetic.

Let us examine how a table would be composed to form addition. At the head of each column would appear one of the numbers to be added; along the side of the rows would appear another list of possible numbers to be added. One would perform addition by finding the column headed by the first number to be added and the row bearing the label of the second number to be added. The sum would be found where these two intersect. Such a procedure could be followed in the adding of small numbers, but think of the size of the tables that would be required to add the numbers used in arriving at the national budget!

The other method of performing arithmetic is to incorporate some rule or procedure for generating the result within the machine. This rule would not depend upon the size of the elements a and b and hence would not impose size restrictions upon the machine.

The Elements Used in Arithmetic

When the elements a, b, and c discussed above are restricted to the integers, the numbers used in counting with signs attached, the computer is called a *digital computer.* When the restriction that the elements be integers does not exist, that is, when the elements used may be "real" numbers or even the complex numbers, the computer is called an *analog computer.* Chapter 7 discusses bases and counting, and the concept of integers is developed there. The above definition is quite precise, but it does not transmit the practical limitations and advantages of each system.

The outstanding quality of the integers is discreteness: there is a big gap between the values of any two consecutive elements which are used by a digital computer. Analog computers use real numbers which have the property of continuity: between any two elements it is always possible to find another element. There is no guarantee, of course, that the analog computer can distinguish this intermediate element from its neighbors.

As an example, consider the input to a computer to be a shaft rotation. The angular position would be read by the computer as one of the elements a, b, or c above. For an analog computer the shaft could rotate to any angular position without restriction and hence could occupy an infinite number of positions. For a digital computer the shaft could stop only at discrete (distinct) positions. This input would be similar to a rotary switch with positive detents at a number of positions so that the shaft can assume only one of the positions for which there is a detent. Increasing the range of the digital computer would consist of increasing the number of switch positions for one rotation of the shaft or of gearing several shafts together. No matter how much the number of positions was increased, these positions or position combinations would still be countable and not infinite.

Editing—What Is It?

One phase of editing is the removal or addition of information to data. For instance, the computer might deal with an employee's pay of $40.00 as 0004000. Before this is written out in the final document, the initial three 0's must be removed and a decimal point inserted between the 40 and the 00. Similarly, it may be desired to print out the date of transaction, even though this was not included in the input data to the computer. This information may be entered into the computer once each day and then printed out with the other information as desired.

The second phase of editing is translation. Information is often handled within the computer in the form of code numbers or letters. Thus, an item in an inventory problem may be referred to in the data-handling operation of the computer by a part number. When the computer writes out information about this part, it must refer to it by name rather than by number. Similarly, during the calculation of the payroll of an employee, he may be referred to by clock number rather than by his name. In writing out his paycheck, it is necessary to refer to the employee by his name. The computer does this task of interpreting as part of the editing.

1.4. WHAT JOBS ARE COMPUTERS USED FOR?

In Section 1.2 a computer's functions have been discussed. In this section we will classify computer applications, some of which were mentioned earlier.

Commercial

High-speed computers can be used for all kinds of accounting and bookkeeping applications; notable among these are accounts receivable, accounts payable, payroll, and inventory. In these applications it is customary to keep the full records within the computer. At any time, at the accountant's discretion, a status report can be made giving the present standing of the company. During normal operation periodic reports are produced for the various accounting and executive branches of the company.

Scientific

Any engineering or scientific enterprise runs across many applied mathematical computational problems, such as the solving of differential