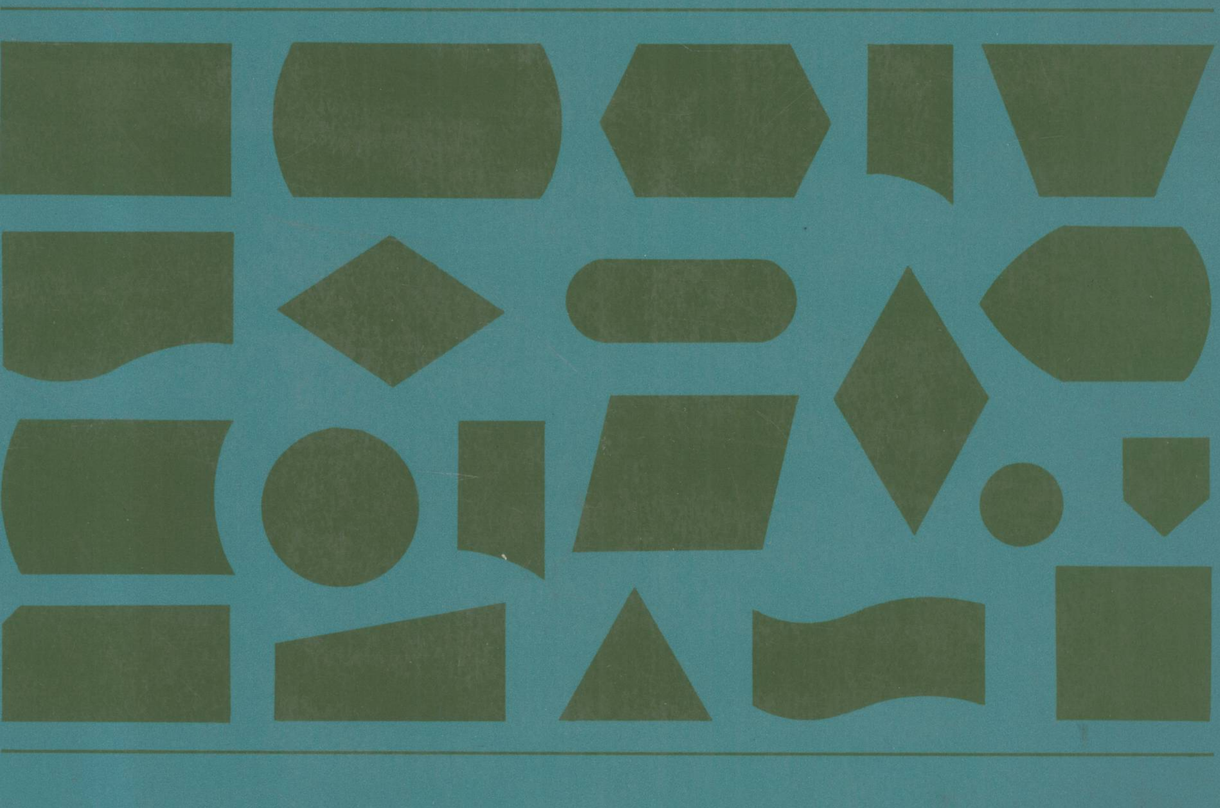
A COURSE ON PROGRAMMING IN FORTRAN



HUNG HING SUM · LOH SHIU CHANG

A COURSE ON PROGRAMMING IN FORTRAN

Hung Hing Sum · Loh Shiu Chang

The Chinese University Press
Hong Kong

International Standard Book Number: 962-201-244-2

First edition 1978
Second edition 1981
Third printing 1982

Typesetting by The Chinese University Press, Hong Kong
Printed by Caritas Printing Training Centre, Hong Kong

A COURSE ON PROGRAMMING IN FORTRAN

Preface

FORTRAN has traditionally been taught in introductory computer courses. This practice has been due in part to the widespread availability of efficient FORTRAN compilers, the problem-solving capability of the FORTRAN language, and the position of FORTRAN as the earliest higher-level language in general use.

There are perhaps hundreds of FORTRAN texts already published through the years: Why another one? The purpose of this book is to introduce the beginning students to the principles essential to an understanding of computers and the discipline of computer programming via FORTRAN. In particular, this text is to respond directly to the needs of students in Hong Kong. Many introductory FORTRAN texts currently available generally suffer from two defects: they do not have sufficient problems to give the beginning student the practice he needs, and the exercises too often assume a firm mathematical background. In our text no knowledge in Computer Science, and a minimum background in mathematics is assumed. It is expected that the text will appeal to students who will take FORTRAN as a formal class or for self-study.

The text is written based on the lecture notes and problems that have been in use in introductory FORTRAN course at The Chinese University of Hong Kong for the last several years. All the examples and programs have been student tested with appropriate degrees of difficulty. In terms of organization, this book has been written with the conviction that the basic prerequisite for an understandable programming text is the logical arrangement and clear exposition of subject matter. This concern with organization is most evident with the orderly presentation of the chapters in this text, which centres upon the step-by-step development of a comprehensive and practical sequence of the art of computer programming. Throughout this text the exposition of each particular topic and FORTRAN programming concept is directly related to the level of difficulty which in our experience the average student is likely to encounter. Simplicity in these instances is correlated with comprehensiveness, not brevity. A three-step development of basic analytical tools is employed: (1) verbal description and illustration, (2) examples and (3) graphic presentation via logic flow-charts based on the examples.

All the concepts presented in this book are illustrated with examples (over 200). The extensive use of examples is consistent with our belief, which stems from our perception of the performance of students in introductory Computer Science courses, that "A well designed example is worth a thousand words." The authors are convinced that the mastery of FORTRAN programming may only be achieved through constant practice. The organization of this text allows the student to write complete and meaningful programs almost from the very beginning.

Chapters one and two introduce the student to a basic knowledge of computers, peripheral equipments and computer programming. Chapters three through eleven present ANSI Standard FORTRAN. Each FORTRAN statement is thoroughly explained and illustrated with at least one example. At the end of each chapter a summary of the General Format of the statements and discussed and more examples are presented. The objective for the inclusion of a glossary in the appendix is two-fold: it serves (1) as a supplement to the first two chapters and (2) as a quick reference in terms of terminology used in Computer Science/Data Processing.

At the end of each chapter, we provide numerous questions and student exercises. End-of-chapter questions and exercises provide pointed recapitulation of each chapter. Much thought has gone into the end-of-chapter questions and exercises. Though purposely intermixed, these questions and exercises are of two general types. Some are designed to highlight the main points of each chapter. Others are open-ended exercises for the more curious and conscientious student. Wherever pertinent, programming problems which require the student to derive and manipulate key concepts and relationships in FORTRAN programming are employed.

The exercises are designed to provide both breadth and depth for the ambitious student. Yet care has been taken to see that they are not beyond the grasp of the average student.

This endeavour has benefited greatly from the advice and counsel of many authors, teachers and students too numerous to express our gratitude individually. The constructive suggestions of many were incorporated into this volume feasible and appropriate.

The authors would like to state their debt to the reviewer of the original manuscript: Mr. Dent-Young. We owe a special debt to Dr. Stanislaus Hu who contributed much to both the content and organization of this text and Mr. Joseph Tu who prepared all the flowcharts and checked the correctness of all the programs. The authors would also like to thank the entire staff of the Computer Science Department at the Chinese University of Hong Kong for their assistance and co-operation.

While we assume full responsibility for any errors of omission or commission, the authors and publisher would sincerely appreciate receiving notices of any annoying typographical errors.

Finally, our families bore patiently the pressures caused by our commitment to complete this book. To all our loved ones, we are most grateful.

June, 1978
Hung Hing-sum
Loh Shiu-chang
Hong Kong

In the present edition all the printing errors have been corrected. The problems in the exercises have been carefully reviewed and amended to provide adequate practice for the students to master the language. New terms have been added to the Appendix to extend the coverage of the jargon.

We wish to express our gratitude to the staff members of the Department of Computer Science at The Chinese University of Hong Kong and particularly to Mr. Joseph Tu and Mr. L. Kong for providing us with some valuable comments and suggestions.

October, 1980
Hung Hing-sum
Loh Shiu-chang
Hong Kong

Table of Contents

Chapter 1 Introduction to Computers

Chapter 1 Introduction to Computers

Although many students quickly learn how to use a computer, they actually have little knowledge about these modern machines. In this chapter we present a brief historical development of computers and introduce some important computer concepts and terminology. We feel that a general knowledge about computers will help the student judge what can actually be accomplished with the aid of these machines.

1.1. A BRIEF HISTORY OF COMPUTERS

The rapidity of the development of computers in the last two decades is a startling example of what can happen when scientific knowledge and technological capacity are both ripe and when the needs of society provide a spur to both. But this development, like all advances in science, has a long history behind it.

The Abacus (Ancient)

The abacus was among the first devices used for computation. It consists of a series of beads strung on rods that are set in a rectangular frame. A horizontal bar divides the beads into two groups. Each bead in an upper group represents five units, and each bead in the lower group represents one unit. The rods represent place values, such as tens and hundreds. To register a value on an abacus, certain beads on each rod are punched next to the horizontal bar. When the value to be represented on a certain rod exceeds 9, the next value (10) is represented by carrying 1 to the next rod on the left and returning the beads on the initial rod to a position away from the horizontal bar. This represents zero. The abacus can be used for addition, subtraction, multiplication (repeated additions), and division (repeated subtractions). It is an example of a digital computing device.

Logarithms and the Slide Rule (1614, 1630)

The slide rule is an example of an analog computing device. Numbers are represented by distances on a scale, and the distances are determined by the logarithms of the numbers. As an example, multiplication of two numbers is carried out by adding their distances on the scale.

Mechanical Calculating Machines (1642, 1672)

Blaise Pascal, a French philosopher and mathematician, developed the world's first mechanical adding machine. In his machine, gears with each tooth representing one of the digits from 0 through 9, were connected in a series. When a gear was rotated past the tooth representing the digit 9, the next gear to the left was automatically rotated one tooth (or digit). In other words, the first gear exchanged ten teeth for one tooth on the next gear. This was the first counting machine capable of carrying a group of ten to the next position. Since the gears could be rotated in either direction, the machine could, in principle, be used to subtract as well as to add.

By 1672 Gottfried Leibnitz, a German mathematician, improved Pascal's machine. He designed a calculator that could multiply by repeated additions and divide by repeated subtractions. The work of Pascal and Leibnitz illustrates the first great idea in the history of automatic digital computation: the recognition that arithmetic operations can be mechanized.

Babbage's Analytical Engine (1812-1834)

The first major step in the development of computers was taken by Charles Babbage, a professor of mathematics at Cambridge University in England. In 1833 Babbage developed plans for an analytic machine. The machine had a store for retaining numbers; a mill that served as a central arithmetic unit for calculating; and an operator to direct the operation of the machine. The sequencing of operations was to be controlled by punched cards. Results would be output on copper plates, and answers could be automatically fed back into the machine so that additional calculations could be performed without further action by the operator. The concept of branching was included in the design simply by advancing or backing up a certain number of cards. Numbers were to be accurate to fifty places, while the machine's memory unit would hold 50,000 digits. The speed of the machine was indicated by the specification that addition could be performed in one second, and a 50-by-50 digit product would take approximately one minute. Unfortunately, the machine was never completed, because his idea was beyond the technical capabilities of that time. Nevertheless, Babbage deserves credit for introducing the idea of data storage, sequential control through programming, and automatic readout—principles that are basic to modern computers.

Boolean Algebra (1854)

George Boole, an English mathematician, developed a system for representing logical statements in terms of mathematical symbols. Using the symbols and some rules one could determine whether a statement was logically true or false. His method was not widely accepted, and it was not until the next century that his ideas were applied in the design of computers.

The Machine-Readable Punched Card (1880-1890)

In 1886 when Herman Hollerith, a statistician and inventor, was working on information in the United States Census of 1880, he developed a system for representing data on punched cards. The machines that Hollerith developed for these cards used electricity to sense the holes in a card and in this manner could count the data from the census. Later, others developed machines which could reproduce punched cards, sort, collate, and even tabulate data. Removable plug-boards controlled the function of these machines. The task performed by a particular machine depended upon the way wires were connected into the plug-board.

Switching Logic (1938)

Claudes Shannon applied Boolean algebra to systematic representation of complex switching networks. Shannon's results simplified teaching and research in the design of circuits like those to be found in future electronic computers.

The Mark I (1937-1944)

Beginning in 1937, Harvard Professor Howard Aiken set out to build an automatic calculating machine that would combine established technology with the punched cards. With the help of graduate students and IBM engineers, the project was completed in 1944. The completed device was known as the Mark I digital computer. Internal operations were controlled automatically with electromagnetic relays; arithmetic counters were mechanical. The Mark I was thus not an electronic computer but was rather an electromechanical one. The Mark I was capable of a precision factor of twenty-three places, and in addition to performing the four basic arithmetic operations, it could reference tables containing previously computed results. Addition and subtraction took exactly 0.3 second; multiplication required about 4 seconds; and a maximum of 16 seconds was required for division. The Mark I weighed about 5 tons and measured 51 feet in length, 8 feet in height. In many respects the Mark I was the realization of Babbage's dream and was the immediate predecessor of automatic electronic computers.

The ENIAC (1943-1945)

The Electronic Numerical Integrator And Calculator (ENIAC) was designed by J. Presper Eckert and John W. Mauchly of the Moore School of Electrical Engineering of the University of Pennsylvannia, and was completed in 1945. The ENIAC weighed approximately 30 tons, took up approximately 1500 square feet of floor space, and used 18,000 vacuum tubes instead of electromagnetic relays as used in the Mark I. With the replacement of the slow electromagnetic relays by vacuum tubes, the ENIAC could process 300 multiplications per second. It could do in one day what would have taken 300 days to perform manually. However, the machine had no internal storage. It had to receive its instructions externally via switches and plugs, a very serious limitation. The ENIAC was used mainly for calculating ballistics tables. It is often identified as the first electronic computer, but is probably more properly thought of as a transition machine.

Stored-Program Computers (1945-1951)

In 1945, John von Neumann, a well known mathematician from the Institute for Advanced Study at Princeton University suggested in a paper that (1) binary numbering systems be used in building computers and (2) computer instructions as well as the data being manipulated could be stored internally in the machine. These suggestions became a basic part of the philosophy of computer design. The first computer to incorporate these ideas was the EDSAC (Electronic Delay-Storage Automatic Computer). It was completed in 1949 in Cambridge, England. EDSAC's instructions were stored internally and were referred to as a program; thus the stored-program computer was introduced. In 1952 members of a development team at the University of Pennsylvania completed another stored-program computer, the EDVAC (Electronic Discrete Variable Automatic Computer). The first electronic computer which was available commercially was the UNIVAC I (Universal Automatic Computer). Unlike its predecessors, it was used for nonscientific data processing. The UNIVAC I was built by Remington Rand in 1949. It used magnetic tape for input and output. Previously, computers had used the considerably slower punched cards and paper tape. Also, the UNIVAC I was the first computer to accept and process alphabetic data as well as numeric data. The first UNIVAC I computer was installed at the United States Bureau of Census in 1951.

1.2. RECENT DEVELOPMENTS

The electronic computer has shown steady progress through several generations. The implication of the use of the term generation is that a significant change in the design of a computer distinguishes it from computers of a preceding generation.

The first generation of computers lasted from 1951 to 1959. The computers of this generation used vacuum tubes, were large, required much air-conditioning and consumed a great amount of power. They were relatively slow; their speed was expressed in milliseconds. In first generation computers the main storage capacity was quite small (2000-4000 words). They had been designed for scientific uses.

The second generation of computers lasted from 1959 to 1964. The computers of this generation were characterized by the use of transistors instead of vacuum tubes. The reason for changing from vacuum tubes to transistors was that transistors are smaller, less expensive, generate almost no heat, and require little power. Thus second-generation computers were substantially smaller, required less power to operate and were more reliable than first generation computers. The speed of second generation computers was measured in microseconds. With second generation computers, internal storage, that exceeded 30,000 words was available. Unlike earlier machines, second generation computers were designed with the needs of business data processing in mind.

The third generation of computers lasted from 1964 to 1970. The computers of this generation were characterized by the use of integrated circuits. The major advantage of integrated circuits is that they are smaller than transistor circuits. They therefore can process data within a smaller area and transfer it more rapidly. An integrated circuit can do the work of a transistor circuit that is hundreds of times larger. Integrated circuits have made possible a reduction in size of computers. Meanwhile minicomputers as well as larger, high-speed computers were made. Third generation computers have internal operating speeds measured in nano seconds and can store hundreds of thousands of words in main storage. The prices of third generation machines are generally lower than those of comparable second generation machines. Another distinguishing factor of third generation computers is that while first and second generation computers were designed to process either scientific or business problems, most third generation computers are general purpose computers in that they are designed to process both scientific-oriented and business-oriented problems.

The fourth generation of computers began in the decade of the seventies. The computers of this generation make extended use of microelectronic concepts to achieve further circuit packing densities. The large scale integration (LSI) of circuits has been developed. Microprocessor based computers are being produced. Semiconductor memory and bubble memory are now being used. A million or more words can be stored internally in the fourth generation computers which are operationally faster and less expensive than the third generation machines.

Research scientists continue to make advances in computer design. Their objective is to develop an even better, more versatile, more powerful computer—one that will operate faster, store more information, require less power, occupy less space and offer greater reliability at lower cost.

1.3. CHARACTERISTICS OF A COMPUTER

The development of computers was discussed in the preceding section. In fact the term computer has been used quite a number of times, but it has not been defined. Before beginning a detailed discussion of the computer, it is important that the key implications of the use of this term be discussed.

Strictly speaking, a computer is any calculating device. The name is derived from the Latin computare, meaning to reckon or compute, and it can as properly be applied to an abacus or a desk calculator as to the modern computer. However, the term "computer" has come to mean a special type of calculating device having the following characteristics.

Electronic: A computer achieves its results through the movement of electronic impulses rather than the physical movement of internal parts. The speed of an electronic computer is the result of electronic circuitry, the speed of which is limited only by the speed at which electricity is transmitted (which is the speed of light).

Internal storage: An electronic computer can hold data and instructions in an electronic representation in an internal memory unit. This feature not only adds to the speed of processing but also forms the basis for a stored program. The speeds associated with processing are measured in microseconds (10^{-6} second) or nanoseconds (10^{-9} second).

Stored program: Data is processed according to instructions. The instructions can be arithmetic, logical and input/output. A detailed sequence, or a number of related sequences of instructions for a task to be performed on a computer is called a program. Both the data and program must be in the storage during processing. The stored program may allow the computer to select a branch of instructions to follow from several alternate sequences. The program may also

allow the computer to repeat or modify instructions as required.
Automatic: The computer is automatic in the sense that the stored program will direct the performance of a sequence of operations without human intervention.

In summary, a computer is an automatic electronic machine with high speed performance, possessing internal storage capabilities, a stored program of instructions, and the capability for modification of the set of instructions during execution of the program.

1.4. THE CLASSIFICATION OF COMPUTERS

Many thousands of computers are in operation to-day, performing a wide variety of functions. There are many ways to classify them. Among these are classification by the type of data they are capable of handling (digital and analog), by the purpose for which they are designed (special and general).

Digital and Analog Computers

There are basically two types of computers—digital computers and analog computers. A digital computer operates by counting numbers. It operates directly on numbers expressed as digits in the familiar decimal system or some other numbering systems. It takes input and gives output in the form of numbers, letters, and special characters represented by holes in punched cards, spots on magnetic tapes, printing on paper, and so on. Digital computers can achieve a high degree of precision. They are generally used for both business data processing and scientific purposes. Most modern computers are digital computers, and it is usually digital computers that are referred to when the word "Computer" is used.

The analog computer, in contrast to the digital computer, does not operate directly with numbers; rather, it measures continuous physical magnitudes (e.g., pressure, temperature, voltage, current, shaft rotations, length), which represent, or are analogous to, the numbers under consideration. The automobile speedometer, for example, is an analog computer which converts drive-shaft rotational motion into a numerical indication by the speedometer pointer. Analog computers are used for scientific, engineering, and process control purposes. For instance, an analog computer may adjust a valve to control the flow of fluid through a pipe, or it may adjust a temperature setting to control the temperature in an oven. A digital computer possesses greater accuracy than an analog computer, but the analog computer can process data faster than a digital computer.

In certain situations (e.g., to similate a guided missile system or a new aircraft design), desirable features of analog and digital machines are combined to create a hybrid computer.

Special and General Purpose Computers

Computers may be designed for either special or general uses. A special purpose computer, as the name implies, is designed to perform one specific task. The program of instructions is built into the machine. This "specializing" of the machine leads to efficient, effective performance of a specific task. A disadvantage, however, is that the machine lacks versatility; it is inflexible and cannot be used to perform other operations. For example, the special purpose computers designed for the sole purpose of controlling a petroleum refinery cannot be used for other purposes without major changes.

A general purpose computer, as the name implies is designed to perform a variety of tasks. This capability is a result of its ability to store different programs of instructions in its internal storage. In short, the stored program concept makes the machine a general purpose device—one that has the versatility to make possible the processing of a payroll one minute and an inventory control application the next. Unfortunately, the ability to perform a variety of tasks is often achieved at the expense of certain aspects of speed and efficiency of performance. In most situations, the computer's flexibility, with respect to its being able to perform a variety of tasks, makes this compromise an acceptable one.

1.5. COMPUTER APPLICATIONS

Computers are now frequently used for both scientific and business applications. Experience has shown that operations most suitable for computer applications are those that have one or more of the following characteristics.

Large volume of input: The greater the volume of data that must be processed, the more economical computer processing becomes relative to other possible methods.

Repetition of work: Much time, effort, and cost are involved in preparing a task for computer processing, it is frequently most economical to use the computer for repetitive projects.

Desired and necessary greater speed in processing: The greater the need for immediate access to information or for rapid turnaround (processing input to produce output), the greater will be the value of a computer relative to slower methods.

Desired and necessary greater accuracy: Computer processing will achieve a high degree of accuracy, if the task to be performed has been properly prepared.

Complex calculations: In some situations calculations of great complexity are involved, there is no alternative to the computer.

1.6. BASIC COMPONENTS OF A COMPUTER SYSTEM

A computer consists of five major components, each of which contributes its part toward the operation of the entire computer system. Fig. 1-1 shows a schematic diagram of the components and how they are linked together. As can be seen, the components are:

The control unit: the central controller for the entire system.

The memory: the storage area for programs and data being processed.

The arithmetic and logic unit: the component in which arithmetic and logical operations are carried out.
The input unit: the part of the computer that accepts information, both instruction and data, and stores it in the memory.
The output unit: the part of the computer that receives results from the memory and presents them to the user.

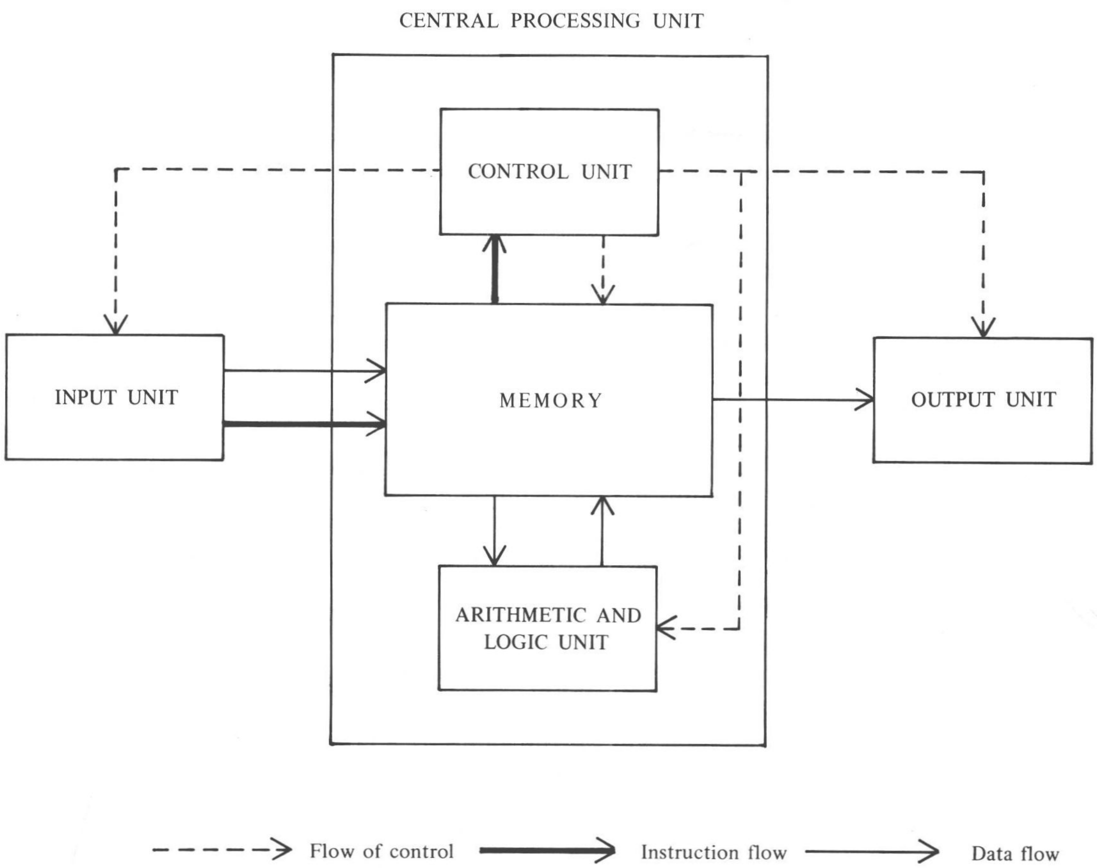


Fig. 1-1. Computer organization

The control unit, the memory and the arithmetic and logic unit are grouped together to form the 'main frame' of the computer system called the central processing unit, or CPU. From an overall point of view, the CPU processes the data transferred to it by the input unit and; in turn, transfers the results of the processing to the output unit.

Let us now consider each of these components in more detail.

The Control Unit

The basic function of the control unit is to supervise the operation of all the