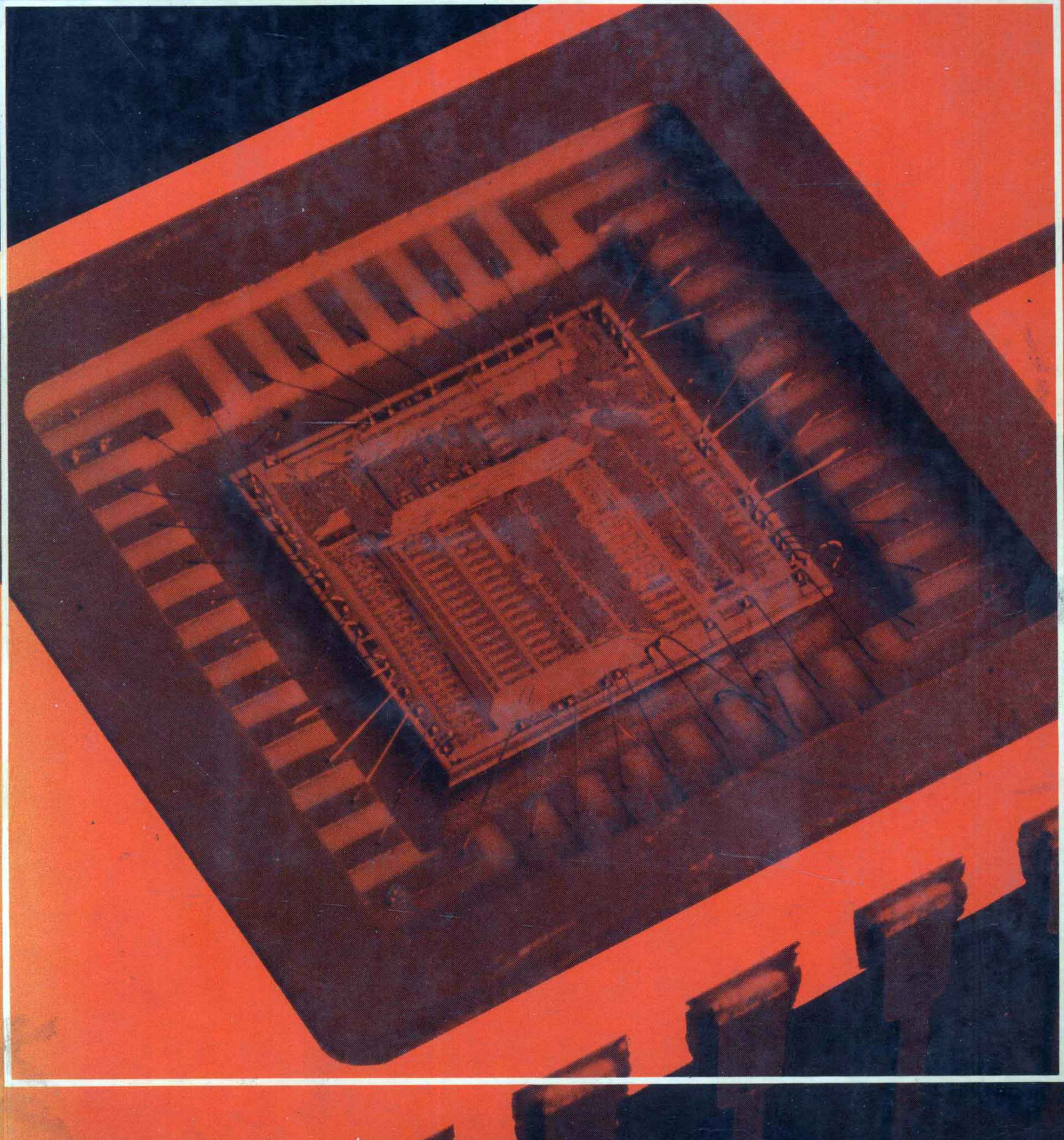
Computing Science

D C Palmer and B D Morris



Computing Science

D C Palmer M Sc, MBCS
Formerly Head of Computing and Mathematics
at St Clement Danes Grammar School
Member of ILEA Advisory Panel on Computing

B D Morris B Sc, FRSA
Lecturer in Computing, Westminster College

Edward Arnold

First published 1980
by Edward Arnold (Publishers) Ltd
41 Bedford Square, London WC1B 3DQ

Reprinted with corrections 1981

British Library Cataloguing in Publication Data

Palmer, D C
Computer science.
1. Electronic data processing
2. Computers
I. Title II. Morris, B D
001.6'4 QA76

ISBN 0-7131-2538-1

To Janet and Marian

Printed and bound in Great Britain by
Butler & Tanner Ltd, Frome and London,

Preface

The authors' intention in this volume is to help readers overcome the most difficult hurdle in computer science, namely getting started on the subject. The contents cover most current 'A' and 'AO' level Computing Science syllabuses and endorsement papers. The book may also be used at 'O' level as a reference for teachers and pupils, and is a suitable introductory volume for a student commencing a more advanced course in computer science.

Chapter 1 gives a brief historical introduction to the computer and its structure. Number base work, ones and twos complement, and floating point arithmetic are dealt with in Chapter 2. Chapter 3 introduces the concepts of Boolean algebra, logic and gates. The treatment is deliberately non-mathematical, and since several different notations are used by the various GCE examining boards, the most popular notations are all introduced.

Problem analysis is covered in Chapter 4, which gives a treatment of a range of topics including flow charts, algorithms, iterative techniques, orders of convergence, sorting, operational research, linear programming, critical path analysis, simulation and other relevant techniques in numerical analysis. The hardware section, Chapter 5, uses the central theme of storage and works from the heart of the computer, i.e. the CPU, out to the peripherals. The principal concepts and devices are dealt with, but it must be realised that the technological development in hardware design is extremely fast. The era of microprocessors is just beginning!

For a full appreciation of Chapter 6, which introduces the analogue computer, it is desirable that the reader should be aquainted with the basic concepts of the calculus. The elements of digital computing are developed in Chapter 7, using simple language ideas devised by the authors to indicate the interaction between control, store, CPU and input/output units. Chapter 8 deals with the principal software techniques, including data structures. Established languages are deliberately *not* used in this section of the book so that general principles can always be applied. However, specific high and low level languages are detailed in the Appendices, together with tutorial advice.

Chapter 9 covers computer systems, e.g. multiprocessing, time sharing and graphic display; this chapter also considers some of the managerial aspects of computing. Some typical applications in a variety of fields are described in Chapter 10, although it will be appreciated that different applications of computing are emerging constantly. Chapter 11 discusses the legal and social aspects of computing and automation.

All chapters include questions designed to test understanding of the text; answers are given as appropriate.

The authors are indebted to their wives for their great encouragement and assistance in the preparation of the book.

1980

DCP
BDM

Acknowledgement

The photographs in Plates 1–7 and in Figs 5.1, 5.20 and 9.7 are included by kind permission of IBM (United Kingdom) Limited.

The description of the examination administration system (pp. 323–4) is included by kind permission of the University of London School Examinations Department.

Contents

1 An introduction to computing

For more than a quarter of a century the computer revolution has been gathering momentum. By the end of 1958, there were 26 main frame computer installations functioning in Great Britain. This had grown to over 1000 installations by 1966, 2000 in 1968 and over 7000 by 1975. The Lindop Report of 1978 envisages that the number of installations will double between 1975 and 1985 but that operational computer installations will then stabilise around that figure. This can be put in perspective internationally when one realises that in 1968 around 72 000 computer installations were in use throughout the world. Computers are being used to carry out an ever-increasing amount of work ranging from clerical drudgery in big organisations to high speed problem solving in research. However, many significant developments in mathematics have been the result of questioning the fundamental rules and axioms which were previously thought to be self-evident. No computer can do this, nor can it use past accumulated knowledge to arrive creatively at new theories and inventions. Computers cannot think: they perform the operations they are instructed to do, i.e. they are just sophisticated machines. The computer has considerably extended the variety of problems that can be tackled, mainly because of the increased speed of calculation. For example, by using a computer two large numbers can be multiplied together in a time of the order of microseconds, which is many million times faster than the pencil and paper technique. Also, increased storage capacity enables several problems to be calculated simultaneously. These capabilities, together with the consistency and versatility of computers, have made it possible for many ambitious projects to be undertaken, including lunar landings.

An introduction to the digital computer

A digital computer is a machine that operates on information, normally in the form of numbers or letters. This information is represented in digital form, usually as ones or zeros, and is termed data. The basic functional components of any digital installation, i.e. its hardware, can be represented in schematic form by a diagram such as Fig. 1.1. The control unit and arithmetic unit are contained in a device called the central processor. Calculations such as addition, subtraction and multiplication are performed in the arithmetic unit. The whole system is governed automatically by the control unit which informs the operator, via the console, of the current state of operations. The console is a typewriter-like device and is used by the computer operator to interrogate the central processor and to instruct it manually if necessary.

Generally, in order to solve a problem, we must have a method of solution and some figures on which to work. Programmers produce the method of solution, or **source program**, written in a high level language; the figures on which the programs operate are called data. Both the program and data are put on an appropriate input medium, e.g. punched cards or paper tape, and read into the computer via an input device. The control

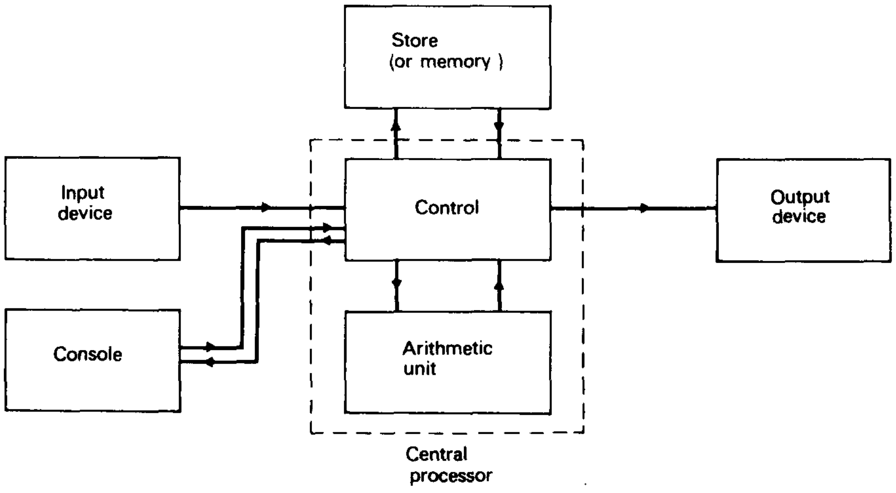


Fig. 1.1 A simple computer structure

unit ensures that the program and data are translated into a form acceptable to the computer, i.e. each instruction is converted into a string of **binary digits**, or bits. Each string of a given number of bits is called a **word**, and each word is then entered into a location in the **store**. The store consists of many such locations, or pigeon holes, each having its own address; when all the instructions are in the store, the source program is compiled into machine code, and when all the data has been fed in, this **object program** can be executed. Data is taken from the store to the arithmetic unit in accordance with the instruction being carried out, and the result of the operation is placed back in the store. This process is continued until the program is fully executed, the answers are then sent to the output device and termination of the job is noted on the console.

Some early developments of computing

Early man, we may suppose, used the fingers of his two hands to represent numbers. The next major step taken by the first civilisations was to represent numbers by means of stones arranged in heaps of ten. This in turn led to the development of the abacus or counting frame which existed in China and elsewhere over 3000 years ago, and which even today is widely used in some countries. By the beginning of the seventeenth century the Arabic system of numeration for both calculation and recording was widespread. To assist calculations in this period, Napier produced in 1614 a device known as Napier's Bones which used the principles of logarithms. In 1624 Briggs published a set of logarithm tables. Later in the century Oughtred developed the first slide rule. During this period Pascal was producing the first mechanical calculating machine. One of the fundamental problems for the machine designer was that of arranging a system to carry from one digital position to the next, more significant, position. Pascal replaced the beads of the abacus by cylinders with the numbers 0 to 9 engraved around the circumference; these were operated by wheels on the front of the calculator, with the carry digits transmitted to the next column by direct gearing of successive shafts. This machine could only be used to add or subtract numbers, but Leibnitz in 1673 completed a machine which could also multiply and divide. The principles involved in the machines of Pascal or Leibnitz are still found in some present-day dial instruments; the gas meter is one example.

The first man to put forward detailed proposals for an automatic calculating machine was Babbage, in the nineteenth century. In 1822 he built his first difference engine to calculate values of polynomial functions; the differences are those obtained between successive values of a polynomial, and are given in Table 1.1 for the function $f(x) = x^2 + x + 3$. The first differences are those between adjacent values of $f(x)$, starting with zero. The second differences are those between adjacent values of successive first differences.

Table 1.1

x	$f(x)$	First differences	Second differences
0	3	2	
1	5	4	2
2	9	6	2
3	15	8	2
4	23	10	2
5	33		

For any polynomial, provided that this process is continued long enough, one of the difference columns will eventually contain the same number in each position. When this constant is found, the machine is set to hold the first value of the function $f(x)$ and all the differences. During each cycle, the next value of the function is obtained by summing the chain of differences and the previous value; all the differences except the constant are updated to their new values by summing each one in turn with those of higher order. The process can be repeated as far as is required.

In the above example, the engine initially holds the first value of the function, the first difference and the constant second difference, i.e. 3 2 2. At the end of the first cycle, the value of the function is given by $3 + 2 = 5$, the first difference is updated to $2 + 2 = 4$, and the engine now holds the numbers 5 4 2. At the end of the second cycle, the value of the function is $5 + 4 = 9$, the first difference is updated to $4 + 2 = 6$, and the machine holds the numbers 9 6 2.

The first model made by Babbage used three numbered wheels to represent the function: two wheels for the first difference and one wheel for the second difference. A larger model giving six figure accuracy was then made, producing 44 successive values per minute. Eventually, Scheutz built a difference engine using fourth differences, with a capacity of fourteen significant figures.

From 1842 to 1848, Babbage worked on the design of a general-purpose digital calculating machine which he called an analytical engine; this had many of the essential features of a current automatic computer. The first use of punched cards was by Jacquard, who built in 1801 a weaving loom in which the movement of threads was controlled by the holes in the cards. Babbage adapted this idea to his analytical engine, with punched cards used for instruction and control. The engine was designed to incorporate a store and arithmetic unit, with 'operations' and 'variables' to be fed in and numbers to be set into the store either manually or from punched cards. Babbage also designed a method for automatic selection of type from boxes to enable the engine to print out its answers; here he anticipated the line printer of the current computer. Babbage's analytical engine represents the transition from a calculating machine, i.e. his difference engine, to the computer. It was unfortunate that his brilliant and original ideas did not get the Government financial backing they deserved for their development. Lady Lovelace, a friend of Babbage, produced many programs using his ideas to perform advanced mathematical calculations. She is considered to have been the first programmer, and her methods were very similar to the machine code language programming of later electronic computers.

Towards the end of the nineteenth century, Hollerith saw that the ideas of Babbage could be modified to produce a machine to speed up the task of taking a national census in the United States. He realised that many census questions are of the 'yes or no' type, and their answers could therefore be represented in a particular position on a card by the presence or absence of a hole, which could be detected by electrical means. The machines Hollerith designed were an outstanding success in quickly processing the data collected for the US census in 1890, in contrast to the twenty year estimate for manual processing. This was the advent of the punched card apparatus from which, in 1911, Powers developed the accounting machines which were the forerunners of those used today.

The early punched card equipment had the limitation that the machine performed only one operation at a time on the given data. This operation could be changed by a plug-board which was attached to the machine, but the stack of cards had to be rerun for each new operation, and there was no facility for forming loops, i.e. returning to the data already handled. These difficulties were initially overcome by the introduction of a facility for performing a limited number of preset operations, and this was the first stage of the stored program.

From the time of Babbage until the late 1930's, the possibility of performing complex arithmetical calculations by completely automatic means was a dream which inspired engineers and mathematicians. However, no large-scale activity in this field was forthcoming until 1936 when Aitken, in collaboration with IBM, started on the construction of his automatic sequence controlled calculator, the Harvard Mark I, which was demonstrated in 1944. Paper tape was used to input the program, and limited facilities for looping were available together with arithmetic operations and functions such as sines and logarithms. Counter wheels were used to store numbers in decimal form in this electro-mechanical machine, and it could store 72 twenty-three digit numbers. Calculation speeds for addition, multiplication and division were 0.3, 4 and 10 seconds respectively. It is worth noting that because of engineering difficulties, this machine did not have a large store or a decision-making facility. ENIAC (Electronic Numerical Integrator and Calculator), built in 1946, was the first completely electronic calculator. This machine incorporated 18 000 valves, and could complete in one hour as many calculations as the Harvard Mark I could do in one week. It could store only twenty numbers, but could add, multiply and divide two numbers in 200, 3000 and 6000 microseconds respectively.

It is considered that the basic structure of a modern general-purpose machine originates from the work of von Neumann together with ideas arising from Turing's theory of decision-making machines. Turing's first work with computers was purely mathematical, and he developed the theory of a universal machine for solving arbitrary mathematical problems. He established that there is a minimum number of steps that a machine requires in order to solve a given problem, i.e. he devised a theoretical method of deciding whether a problem is soluble.

In the middle and late 1940's, groups of engineers and mathematicians at the Universities of Cambridge and Pennsylvania were working on the implementation of von

Neumann's proposals for a machine in which an **internally-stored program** could be used. The first of these automatic general-purpose electronic computers were the EDSAC (Electronic Delay Storage Automatic Calculator), built in Cambridge by Wilkes, and the EDVAC (Electronic Discrete Variable Automatic Computer), built in the US by von Neumann; both computers came into operation in 1949. These machines included many new features, but the most important was the internally-stored program. All the instructions for the computer were stored electronically within the machine, and it could therefore operate at electronic speeds without human or mechanical intervention. Machines called computers have a stored program, and this distinguishes them from those called calculators, which do not. May 6th, 1949, is a landmark in the history of the computer, because on this day EDSAC used a stored program to print out a table of the squares of the numbers from 0 to 99. This historical print-out from the first programmed electronic computer is displayed in the Science Museum, London.

At this time, computers were not sold commercially, and in 1952 J. Lyons and Co. used an EDSAC machine to develop and build LEO (Lyons Electronic Office), the first commercial computer, which was used for cost account analysis. However, in the US, the EDVAC team realised the value of their stored-program computer and persuaded the Remington-Rand Company to market a machine of this type. In 1951, the Remington-Rand Company produced a large stored-program computer under the name of UNIVAC (Universal Automatic Computer) and in 1953, IBM (International Business Machines) entered the field with the IBM 701 machine. In 1959, IBM brought out a further two computers, the IBM 1401 and the IBM 1620, and in 1964 the IBM System 360 was marketed. The IBM company now has the major share of the world computer market, although during the late 1950's many other companies embarked upon the construction of computers, including ICT (later ICL), Burroughs and Honeywell. The speed of operations has steadily improved since the early computers, and the operation time for an IBM 360/50 for the addition of 2 numbers was 20 microseconds and for multiplication 32 microseconds.

Since 1949, there has been considerable progress in the development of **hardware** (machines) and **software** (programs which enable us to use the machines). The most significant advances in hardware technology have been the availability of the transistor, core store, magnetic tape and integrated circuits. Transistors were discovered in 1948 by Shockley, and they soon replaced valves in computers. Their use greatly reduced power consumption and machine size, and increased reliability.

The principle of magnetic core storage in the heart of a computer has proved to be a simple, economic and lasting development. It is a well-known physical fact that a wire carrying an electric current will induce magnetism in an adjacent piece of iron, and the effect is even more pronounced if the wire is threaded through a small core (ring) made from a ferromagnetic material. Core store consists of a few million rings, each threaded in two directions by thin wires to form a robust matrix. Each core can be easily magnetised, and will retain its magnetisation indefinitely if not disturbed. A threshold (minimum) current will reverse the sense of the magnetisation, and the whole unit can be quite compact because very small rings can be manufactured.

The advent of the magnetic tape has greatly enhanced the capacity of the backing storage of the computer: a punched card holds approximately seven characters per inch, while a magnetic tape may contain up to 556 characters per inch. Conventional wiring was soon replaced by large-scale integrated (LSI) circuits, in which components and their interconnections are formed in one unit. The use of such devices greatly improves the speed of operation and reliability of the computer, as well as reducing its size. The lack of wiring contrasts with the Harvard Mark I, which used over 500 miles of wire.

The term 'software' covers the programs which are used to control the operation of a machine. In the 1940's, the software requirements of the first EDVAC and EDSAC computers depended upon the decisive theoretical concept of the stored program, as devised by von Neumann.

The first digital computers were given instructions in machine code, a language which reflects the hardware structure and directly operates the machine. Thus the first programmers required detailed knowledge of the methods of storage and data manipulation within the machine. The early programs were very complicated, and difficult to check and modify, because the movement of every number within the machine had to be specified precisely. The basic steps were limited to the machine's instruction set, which consisted of simple operations such as read, write, move from store to register, add, subtract, shift left. These programming languages were called 'low level' because they could be used without interpretation by the computer, and since early program writing was a very skilled operation, many programmers held a degree in mathematics.

The next stage in programming was to introduce words that were interpreted as numbered instructions by the computer; this was called **symbolic coding**.

In 1951, the Cambridge University Mathematical Laboratory published a volume of subroutines. These were commonly-used sections of programs, e.g. numerical operations such as subtraction, division, square roots, read in, print out. Each one was prepared once, stored, and used repeatedly. Autocodes were introduced in the late 1950's, and these enabled one input instruction to bring into operation a complete subroutine. This was the start of the move from machine orientation of programming to problem orientation, and necessitated the development of 'high level' languages using terms which resembled both normal English and mathematical notation. FORTRAN was the first high level language, and this was developed by IBM. A parallel development came in the error diagnosis of programs.

A **compiler** is a special program for translating a high level language into a low level one, and the first compilers were produced in 1957 and 1958. The most important of these was the FORTRAN compiler for the IBM 704, and this took eighteen man-years to write. FORTRAN is a scientifically orientated language, and the first commercially orientated language, COBOL, was developed in 1959. In 1960, an international committee devised a universal language called ALGOL 60, but although it was intended that ALGOL could be used on any computer, this vision has not been completely realised. There are now numerous high level languages, but ALGOL and FORTRAN are among the more popular for scientific work, COBOL is used extensively in commerce, and PL1 is employed in both fields. The appropriate compilers are provided by the computer manufacturer for each of its machines.

Time sharing is the simultaneous interaction with a computer system by several users. This concept was developed by Shaw and Baker of the Rand Corporation, and their pioneer software system for this application was called the JOSS system.

Further advances have been made in the production of software, increasing the sophistication of its use in established fields, widening its applications to more diverse fields and, where possible, simplifying and standardising procedures. Before these developments can be usefully considered, we need to study the computer structure in detail.

The computer revolution

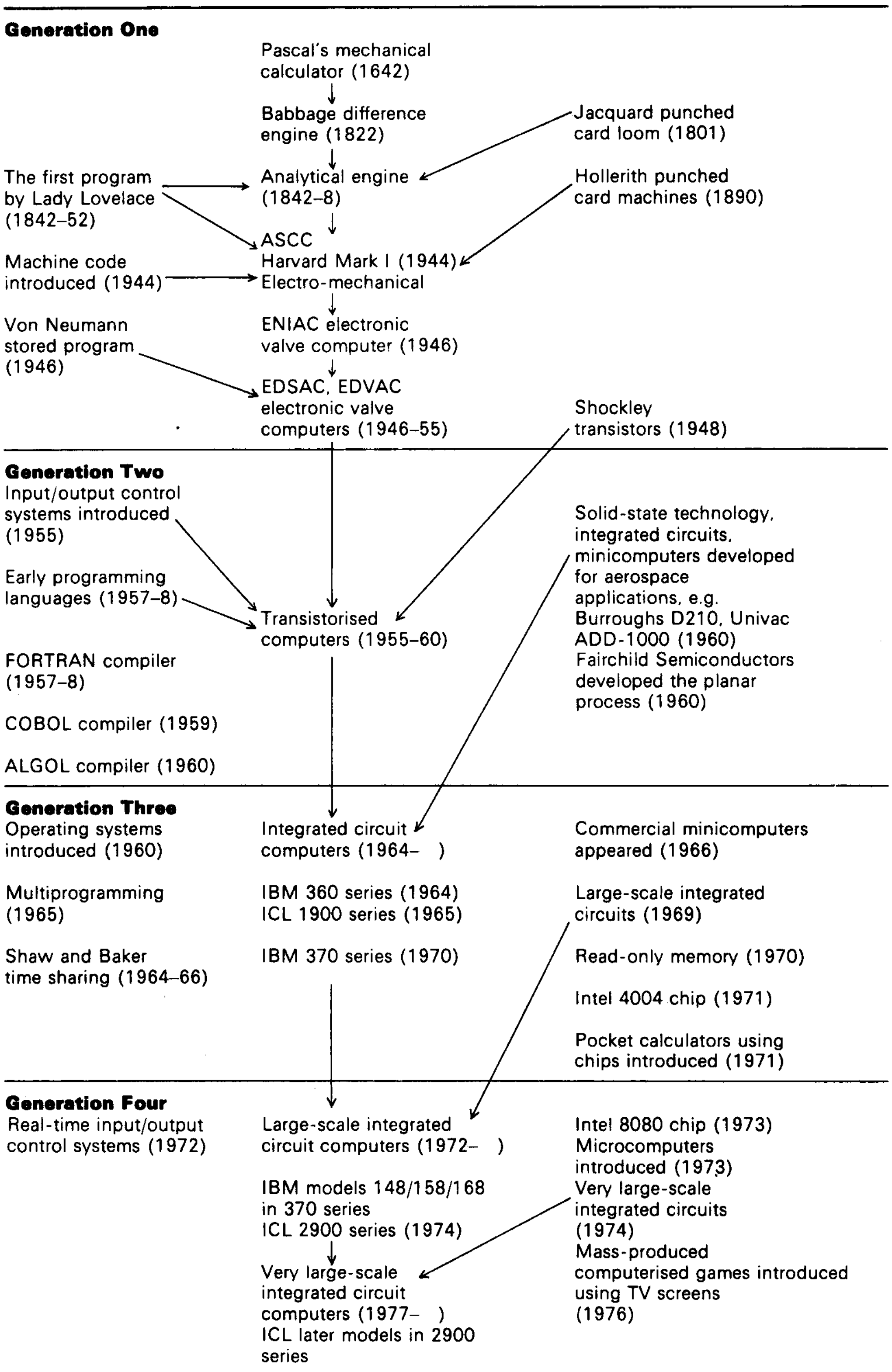
Figure 1.2 and Table 1.2 indicate the basic elements in the development of computing, and as far as possible categorise these elements in generations One to Four with their principal characteristics. It must be appreciated that some computers have mixed generation construction, for example the Honeywell 1250 computer had transistor and integrated circuits intermixed.

Table 1.2 Characteristics of generations of computer systems

Generation	Hardware	Logical organisation	Software
One	Vacuum tubes.	Operations strictly sequential.	Program development, program loader, assembler.
Two	Transistors, magnetic core memories. Increased speed, moderate capacity.	Input/output operation able to proceed simultaneously with calculations. Interrupts.	Input/output control system. High level languages. Compilers.
Three	Integrated circuits. Higher speeds, greater capacity. Minicomputers developed.	Able to handle many programs at the same time. Memory protection. Modular construction introduced.	Operating system introduced, handling efficient operation of input/output devices, interrupts, and taking over large part of operator's previous job function. Multiprogramming, time sharing.
Four	Chips or mass produced processors. Large capacity microcomputer developed. Costs greatly reduced.	Emphasis on modular design. Symbiosis between hardware and operating system.	Large capacity data storage files – laser techniques. Sophisticated graphical displays. Computer-aided instruction. Real-time control system.

The microelectronics industry has its origins in California, USA. It was there that the designs for the first processor on a chip were produced. Most chips at present are made in America by such firms as Intel, Mostek, Texas Instruments, National, Motorola and Fairchild, with the Japanese firms Hitachi and Toshiba making a significant contribution. The British government through the National Enterprise Board has supported former Mostek employees in setting up Inmos, a British company, to design and produce chips in the UK, despite a court case in Dallas. The NEB has also backed Insac and Nexos, two British firms, to develop the use of chips in software and office equipment respectively. The UK has some of the world's best software writers, and Insac should use this to its advantage. Motorola and Mostek have established factories in Scotland and Eire respectively to produce chips.

A minicomputer is structurally a small version of a main frame computer. It is used for low volume applications which require a relatively sophisticated computational capability. A minicomputer system is sold by trained sales and applications engineering staff as a solution to a problem, i.e. manufacturers supply a complete system including peripherals, software, support and maintenance. The earliest minicomputers were developed for aerospace applications and appeared around 1961–2, e.g. the Burroughs D210 and the Univac Add-1000. Commercial minicomputers appeared around 1966 from such manufacturers as Honeywell and Hewlett Packard.


Generation One
Pascal's mechanical calculator (1642)
Babbage difference engine (1822)
Jacquard punched card loom (1801)
The first program by Lady Lovelace (1842–52)
Analytical engine (1842–8)
Hollerith punched card machines (1890)
ASCC Harvard Mark I (1944) Electro-mechanical
Machine code introduced (1944)
Von Neumann stored program (1946)
ENIAC electronic valve computer (1946)
EDSAC, EDVAC electronic valve computers (1946–55)
Shockley transistors (1948)
Generation Two
Input/output control systems introduced (1955)
Early programming languages (1957–8)
Transistorised computers (1955–60)
Solid-state technology, integrated circuits, minicomputers developed for aerospace applications, e.g. Burroughs D210, Univac ADD-1000 (1960) Fairchild Semiconductors developed the planar process (1960)
FORTRAN compiler (1957–8)
COBOL compiler (1959)
ALGOL compiler (1960)
Generation Three
Operating systems introduced (1960)
Integrated circuit computers (1964–)
Commercial minicomputers appeared (1966)
Multiprogramming (1965)
IBM 360 series (1964) ICL 1900 series (1965)
Large-scale integrated circuits (1969)
Shaw and Baker time sharing (1964–66)
IBM 370 series (1970)
Read-only memory (1970)
Intel 4004 chip (1971)
Pocket calculators using chips introduced (1971)
Generation Four
Real-time input/output control systems (1972)
Large-scale integrated circuit computers (1972–)
Intel 8080 chip (1973) Microcomputers introduced (1973)
IBM models 148/158/168 in 370 series ICL 2900 series (1974)
Very large-scale integrated circuits (1974)
Very large-scale integrated circuit computers (1977–) ICL later models in 2900 series
Mass-produced computerised games introduced using TV screens (1976)