The computer in the university

By R. A. Buckingham*

This paper discusses the growing importance of computer science in universities; the functions of a university computing laboratory in developing teaching and research in computer science and in providing a computing service; and the possible effect of these diverse needs on computing systems in the future.†

In discussing the significance of the computer in the university I shall not be much concerned with the nature and characteristics of computers themselves, as these are fortunately beginning to be well understood. I shall be concerned with three aspects of the impact which computers are having on what goes on inside a university, these three aspects being teaching, research and the computing service.

Computer science

I would like to deal first with the impact on the educational work of the university, beginning with a few remarks about the term "computer science" itself. The University of London now has an Institute of Computer Science, at least two departments with this title, and offers a Postgraduate Diploma in Computer Science. There may still be—indeed there are—some doubts about the status to be given to the branch of knowledge which goes under this name. Is it to be regarded as a new discipline, comparable with say engineering or economics, or merely as a hotpotch of technical knowhow connected with computers?

Although my own view inclines towards the first of these, let me say that I do not wish to regard the computer itself as the indispensable core and the main focus of attention of this new subject. This lies perhaps more within the scope of engineering. I do not know of any discipline yet worth the name which depends solely for its existence on a tool. What is significant is the fact that computers exist and are capable of carrying out routine operations at a rate of millions per second, and soon perhaps at hundreds of millions per second. Once this idea and the possibility of its realization had penetrated our conscious thought it was inevitable that new modes of thinking should arise and a new body of knowledge emerge. To quote Dr. Licklider (Licklider, 1962) from a symposium held three years ago at M.I.T., "The impact of the digital computer upon university education will stem mainly from the changes the computer will produce in intellectual activities generally."

It would seem worth repeating here an argument put forward by Louis Fein (Fein, 1961) in a paper entitled, "The computer-related sciences (synnoetics) at a university in the year 1975." This briefly is that we have reached a stage in the development of human society when it is essential, in order to solve the many complex problems of continued existence at the present technological level, that man's own mental powers should be enhanced. Man's physical powers were enhanced by the developments of the industrial revolution; the time is now ripe for an enhancement of his mental powers, which are primarily the ability to invent and to reason. Opportunely he has invented the computer having limited mental capabilities which provide a valuable extension of man's own. It therefore becomes necessary to study the conditions under which a system comprised of "people, mechanisms, plant or animal organisms, and automata" may develop mental powers exceeding those of its components.

This study Fein calls *synnoetics*, and it is one which he regards as supradisciplinary in the same sense as logic and mathematics are supradisciplinary, i.e. the study leads to methods and ideas which may be applied to the solution of problems in many other disciplines.

We may be a long way from forming a comprehensive theory of synnoetic systems but in the meantime the study of man-computer systems, and the search for combinations which take full advantage of the things which men do well and those which computers do well, is very much to the fore. It certainly forms an essential part of what we regard as computer science. The whole field is developing so rapidly that it is still difficult to define in terms of subject matter that may be systematically taught. An attempt to do so was made recently by Professor Keenan of the University of Rochester (Keenan, 1964), and I find it difficult to improve upon his description, which includes four main areas of study. Slightly amended, these are:

1. The organization and interplay of the parts, which may include both machines and people, which make up an information-processing system.

† This paper is based on the author's inaugural lecture given at the University of London on 26 October 1964. The lecture was illustrated by a number of sound recordings of computer-programmed music, a computer musical composition, and of synthetic speech generated by computer; also slides of input and output, and a film of satellite motion produced by computer.

* *Director, Institute of Computer Science, University of London,* 44, *Gordon Square, London, W.C.*1.

2. The development of software systems with which to control and communicate with equipment. These include programming languages, executive systems, and other systems associated with the input and output of information.
3. The study and derivation of procedures and basic theories for specifying processes. This would bring in such subjects as numerical analysis, symbol manipulation, heuristic methods, and theoretical methods of information storage and retrieval.
4. The application of information-processing systems, software systems, and the theory of procedures and processes, in other disciplines.

The first three of these areas define the scope of the basic science; the fourth constitutes the applied science, and by bringing to light potential uses of computers provides an essential stimulus to the computer scientist.

The teaching of these subjects has hitherto been mainly to graduate students who have necessarily been drawn from other fields and who have felt the need to apply the computer in their research or their professional work. We recognize of course that computer science, like mathematics, is the handmaid of many other disciplines, from economics and psychology to physics and engineering, and the techniques of applying it must be taught as one teaches the use of the differential calculus. However, this should mean far more than the mere teaching of how to program a particular computer. At the very least it is important to convey to students the need for the complete and proper formulation of problems, the need to study and understand processes, the use of flow diagrams in describing processes, and the design of algorithms or procedures for use in problem-solving. The present emphasis on all these things is largely a by-product of the development of computers, and they are worthy of inclusion in many courses at undergraduate level. This is indeed happening, for example, in some departments of engineering, and in some U.S. universities on a wider scale. To quote Professor Perlis (Perlis, 1962) "a course in programming, if it is taught properly, is concerned with abstraction: the abstraction of constructing, analyzing and describing processes." As such he thinks it important in a liberal arts course as in science or engineering.

Such teaching may or may not be accompanied by ready access to a computer. I believe it gains immensely in effectiveness if this access is possible; and given a fast, time-sharing machine it is practicable. If 1,000 students each generate 10 computer programs in the course of a term, this means a daily load of 200 programs. If the machine is fast enough, and there are efficient means of transmitting the problems directly to it from a distance, and output is carefully restricted, then the central processor need not be fully engaged for more than 10 or 15 minutes a day. It should also carry out all the essential marking and grading of the students' work.

The time will soon arrive when it is necessary to provide more extended courses in computer science which will form the major part in a first-degree programme. I will not attempt here to outline what these courses may contain; there is no lack of possible material, though a great deal exists only in the journals and has yet to be assembled in a coherent and palatable form. Perhaps it will suffice to say that computer science courses may be of two kinds: one primarily mathematical but strongly oriented towards the theory and use of digital computers, the other mainly concerned with information-processing systems and their design, control and application. In this country a start is about to be made on a three-year course at the University of Manchester.

Manpower requirements

Perhaps a brief comment on the urgency of the educational problem will not be out of place. Surveys have recently been made both in the United States (Brandon, 1964) and in Europe (de Bruijn, Frielink and Scheepmaker, 1963) on the likely requirements for computer specialists, in the fields of systems and problem analysis, programming and operations, by 1970. In the U.S. an addition of at least 160,000 to the present force (of about the same size) by 1970 is estimated; in the countries of Western Europe a total of over 200,000 has been forecast. These figures exclude computer engineers, and the needs of computer manufacturers, and all those for whom computers are only a part-time activity. In this country, on the basis of 4,000 computers in 1970 it would seem that 40,000 specialists will be needed if Britain is not to fall still further behind other European countries, and of these at least half should preferably be graduates. Recruitment from other disciplines is hardly likely to meet all the demand, and an immediate expansion in the opportunities for direct education in computer science in the universities is very necessary.

Research

Let me now turn to the relation between the computer and creative research in general in the university. I said earlier that there were some things which computers are good at, and others which human beings are good at. Amongst the special attributes of the computer are the following:

1. A facility for very rapid calculation, which is valuable both for simple routine operations and for more difficult and abstruse calculations.
2. More generally the manipulation of symbols of all types, whether abstract (in the sense that they do not directly convey information, e.g. numbers, words) or less abstract, such as maps, pictures or patterns which do convey information.
3. An ability to analyse completely, in a reasonable time, a complex structure such as a tree or network in many dimensions.
4. Following on this, the analysis of recursive structures, perhaps involving repeated interruptions to enter a nested sequence of subroutines, some of

which may be identical with routines already entered.

(This is difficult for humans. Professor G. A. Miller of Harvard recently quoted the following sentence of nested clauses: "The film that the script that the novel that the producer whom she thanked discovered became made was applauded by the critics.")

5. The generation and analysis of large numbers of random events within a framework of restrictions appropriate to the problem being investigated.

It is to be expected that in the university setting all these special qualities of the computer should be exploited to the full. The ability for high-speed numerical processing has certainly been applied most extensively, and is to the fore in the large-scale data processing to which experiments with high-energy particle accelerators give rise, in the analysis of records from space satellites, or in the elucidation of complex crystal structures. I need not list the many large computing problems which arise in the study of molecular structures, fluid flow, the design of engineering systems, and the optimization of industrial or economic processes which depend on large numbers of variables—and in many other fields. All these have been widely developed, and if allowed to do so would keep an Atlas computer occupied for the greater part of its time.

The other facilities have not perhaps been fully exploited so far, although there is no reason why they should not be, to an increasing extent. Symbol manipulation in the general sense, and the use of recursive operations, are indispensable to current techniques of compiling programs written in various programming languages into sequences of machine instructions, and to linguistic analyses generally. They could be used more fully than now in algebraic manipulation, symbolic integration, and the like.

The indexing and retrieval of information on a large scale will depend for success partly on developing suitable methods of analysing large tree and network structures. This may also be true of decision-making procedures important in the fields of management and industrial processes. In the same context there is a great deal of interest in heuristic methods which, in problems where the possible number of pathways is excessively large, reduce the number of alternatives to be examined to a practicable level by introducing certain artificial goals or objectives which should be satisfied.

The generation of random processes has already had many applications; from the evaluation of multi-dimensional integrals to the structure of liquids and the behaviour of nuclear reactors. It has also been exploited to a limited extent in design problems of an artistic kind. Two of these which I would like to mention are the generation of geometrical patterns, as illustrated in **Fig. 1**, and musical composition.

Human beings have capabilities which the computer is never likely to match. Our capacity to retain information is still in excess of any computer, although the storage devices promised with some modern computer systems do not fall so far short of the 10^{12} bits of information which Von Neumann estimated to be the capacity of the human brain. We can retrieve much of this information in a remarkably short space of time, though not always with absolute certainty and reliability. Our perceptive abilities, especially with the senses of sight and hearing, assisted by speech, are still unrivalled. We also have a power to make decisions based on incomplete or inadequate evidence which we may, rightly or wrongly, prefer to those of any machine.

No doubt computers will continue to improve in these respects, but the vital difference remains that man, through living and reacting to his environment, is able to attach meaning to the symbols which he uses for purposes of description, reasoning, and planning his life. A simple example will perhaps illustrate this difference. In C. K. Ogden's *Basic English Dictionary* the author has defined 30,000 words in terms of 800. These 800 are the words of Basic English which the author says it is useless to define in terms of others. It is fundamental that an understanding of some words is not acquired in terms of other words but through the process of living. A computer on the other hand can happily manipulate the 30,000 words using the network of relations between them and the basic 800, but when a question is asked about basic meanings it has no recourse to its mother's knee.

An increasing amount of research using computers is concerned with artificial intelligence, and in making a machine learn to carry out certain tasks. I am not convinced that it is feasible or desirable to develop "machines that think"—as a replacement for the human brain it is still decidedly expensive—but it is immensely valuable and worthwhile to simulate human thought processes using a computer.

Computer translation of spoken languages may be another dream which will prove never to be economical except in rather limited circumstances, and is always likely to be circumscribed by the ultimate lack of understanding of meaning which is an inherent limitation of the machine. Despite this, much useful knowledge about linguistic forms has already emerged through computer-oriented research, and a study of languages, syntax and even semantics, is an essential part of computer science.

It is important that work in all these fields should find its place in the university, and that the necessary time and facilities on the computer should always be provided.

There is one further research need to which I should refer, namely the opportunity to attach on-line other equipment, which will transmit information at regular intervals to the computer and expect to receive suitable signals in return. The computer is here part of a closed loop and is taking an active part in an experimental investigation. One example of this is the adaptive control experiments being planned in the Electrical Engineering Department of Imperial College, and for which provision is being made on the London Atlas computer.

```
 1  :SQUARE
BOTTOMLEFTHANDCORNERAT:  10.375     45.002
SIZE=  21.843

 2  :SQUARE
BOTTOMLEFTHANDCORNERAT:  66.714     74.957
SIZE=  36.046

 3  :GENERALTRIANGLE
X1,Y1=  74.276     96.140
X2,Y2=    70.964     51.358
X3,Y3=  18.615     57.933

 4  :GENERALTRIANGLE
X1,Y1=  19.413     87.871
X2,Y2=   19.262      8.437
X3,Y3=  38.374     47.441

 5  :GENERALTRIANGLE
X1,Y1=  26.418     97.457
X2,Y2=    53.196     60.298
X3,Y3=  13.171     83.437

 6  :SQUARE
BOTTOMLEFTHANDCORNERAT:  46.598     65.219
SIZE=  32.411
```

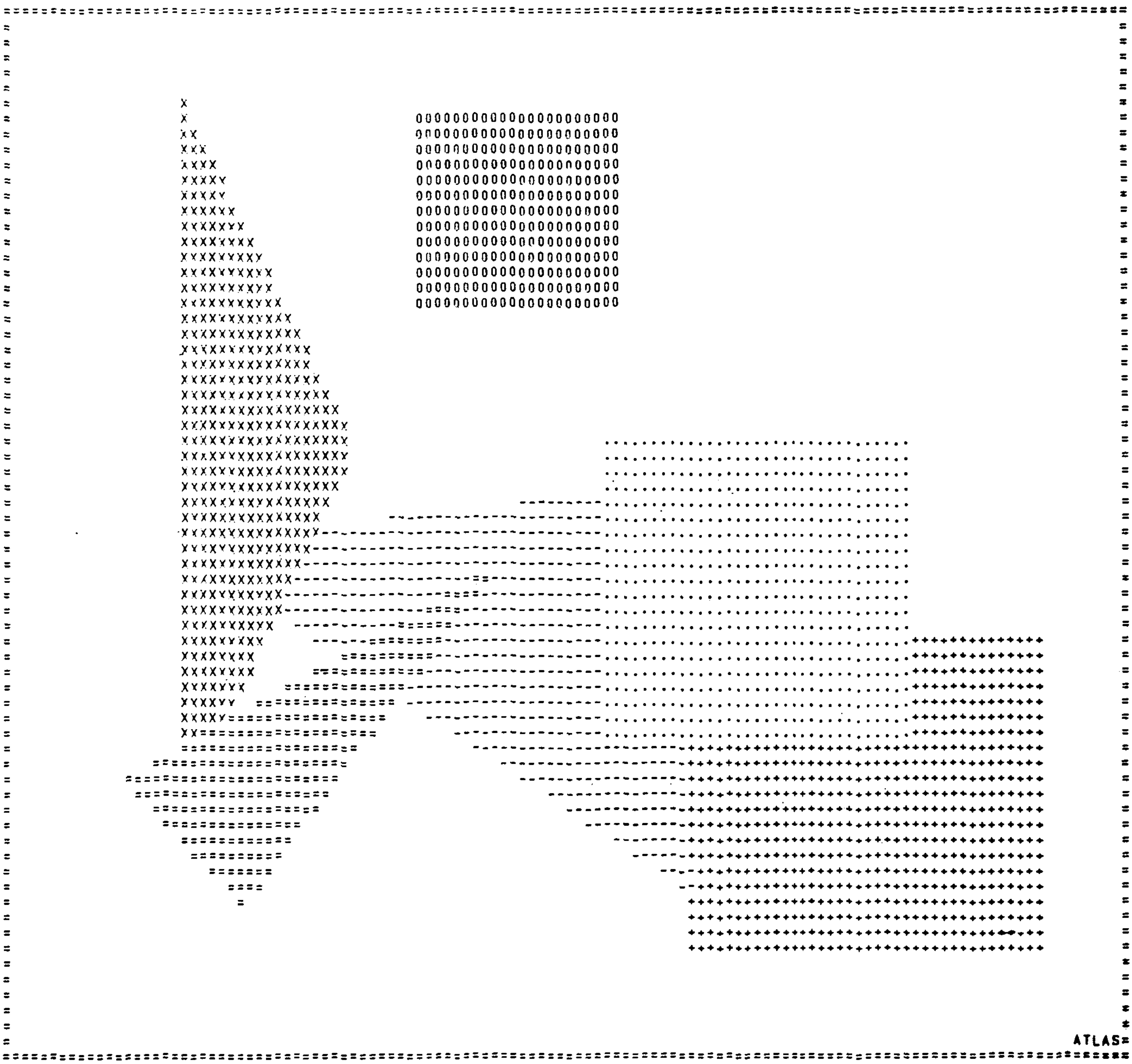

Fig. 1—Sample output from geometrical design program (*by courtesy of A. J. T. Colin*)

Computing service

Let us now consider how the requirements for the teaching of computer science, and the computing needs of university research affect the computing service to be provided. I have first wished to create and justify the impression—I hope successfully—that in setting up the Institute of Computer Science as an entity apart from the Atlas Computing Service, the University of London has bravely foreseen the needs of teaching and research in computer science, and has recognized that these should not be confused with the organization of computing facilities.

These are important, and of course very closely related to the activities of the Institute. It should be evident that in a university setting the requirements to be satisfied by a centralized computing service are extremely varied. They include the following:

(*a*) The facility to develop many programs, large and small, on the computer, and to provide the authors of them with a rapid turn-round, often of the order of minutes, and preferably never more than one or two hours.

(Some 10–15% of the university load in respect of computer time is concerned with program development and testing; in number of runs the percentage may be much greater.)

(*b*) An adequate advisory service to programmers regarding the use of the programming languages available, and assistance in piloting their programs through the system.

(*c*) The maintenance of several major and efficient programming languages, and perhaps some special-purpose languages also, including aids to fault-finding in programs, and all the relevant documentation.

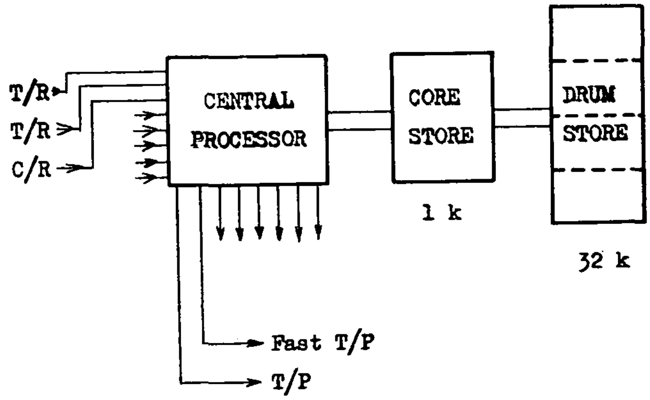

Fig. 2—Simple Mercury system

(*d*) Standard programs for a variety of routine processes, e.g. statistical analysis, survey analysis, matrix algebra, and so on, and a comprehensive library of other programs to carry out simple procedures efficiently.

(*e*) Facilities for large-scale data processing.

(*f*) The handling of large numbers of student programs.

(*g*) Opportunity to develop and exploit new computer techniques.

To the above one should add, though this may be more controversial,

(*h*) Facilities for on-line working with experimental apparatus.

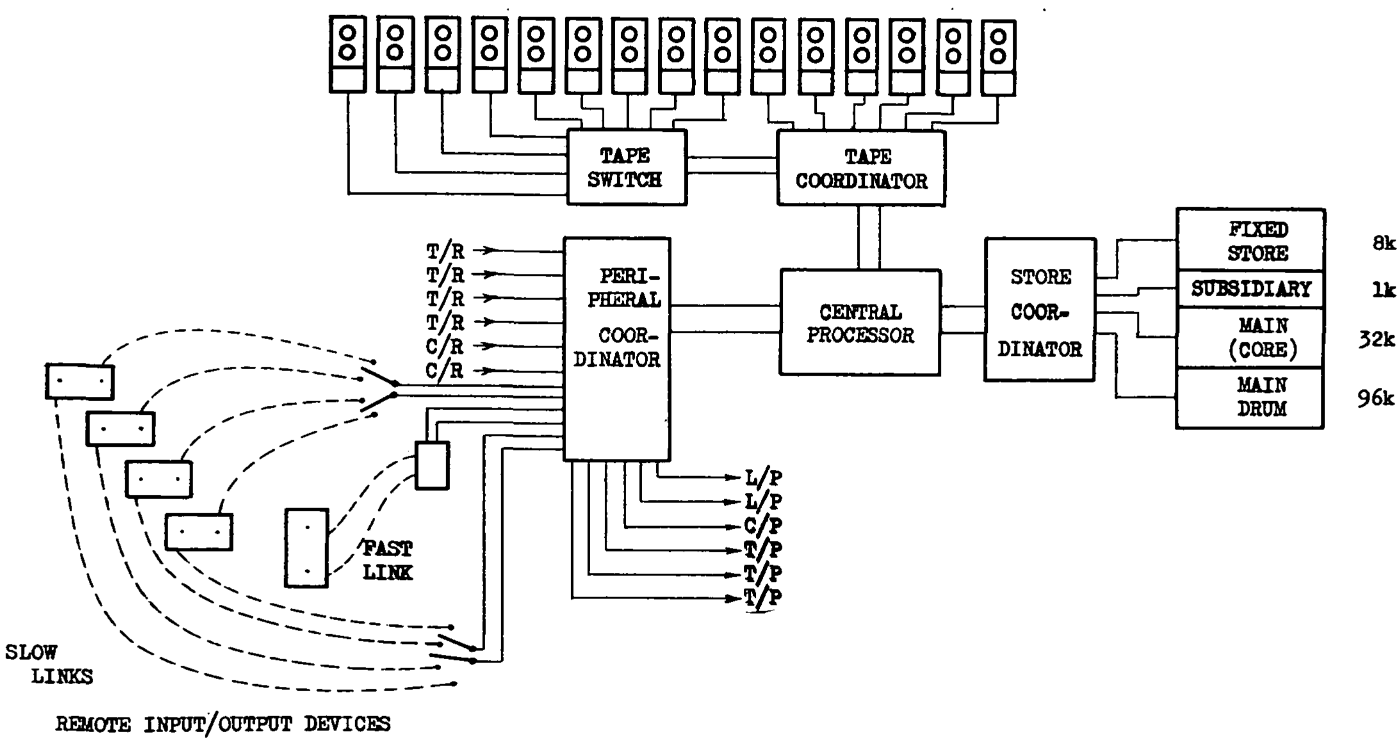

Fig. 3—Atlas 1 system

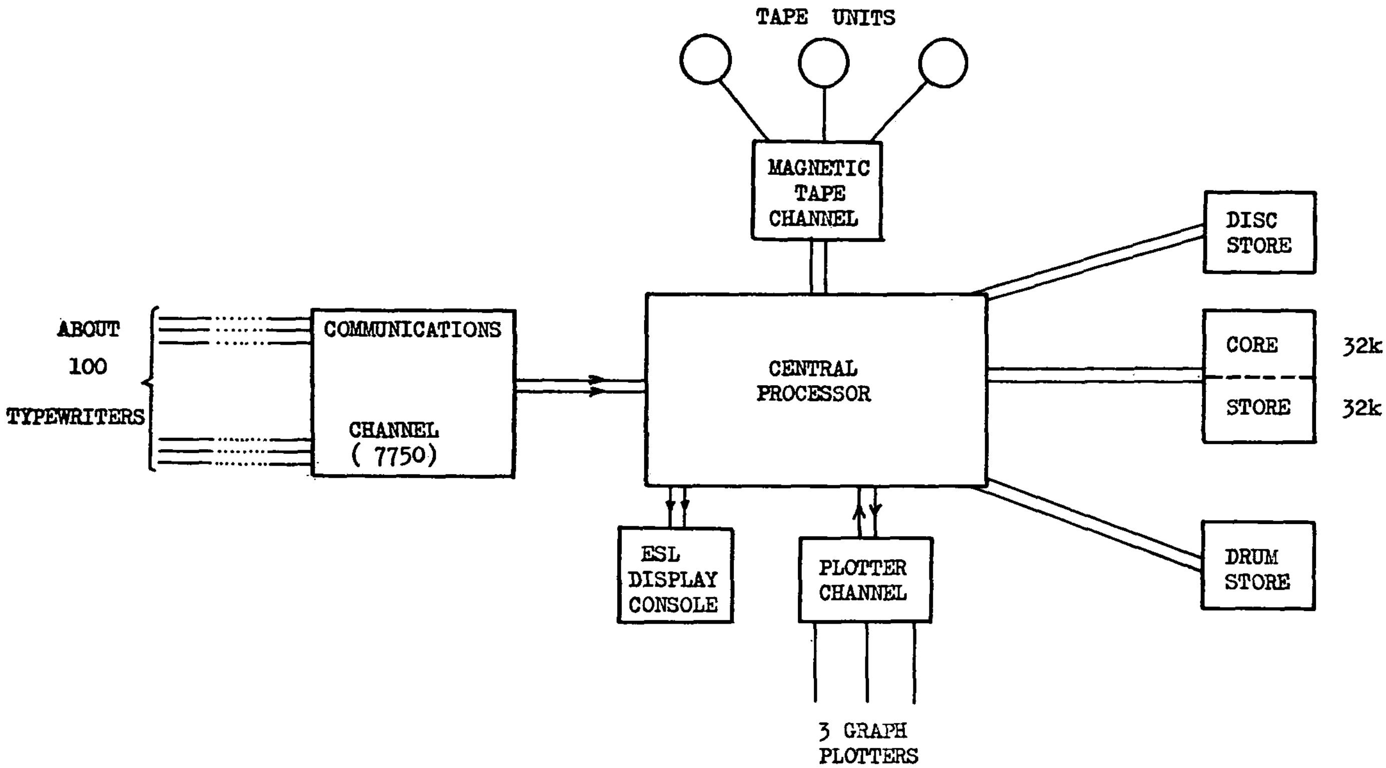

Fig. 4—Project MAC system

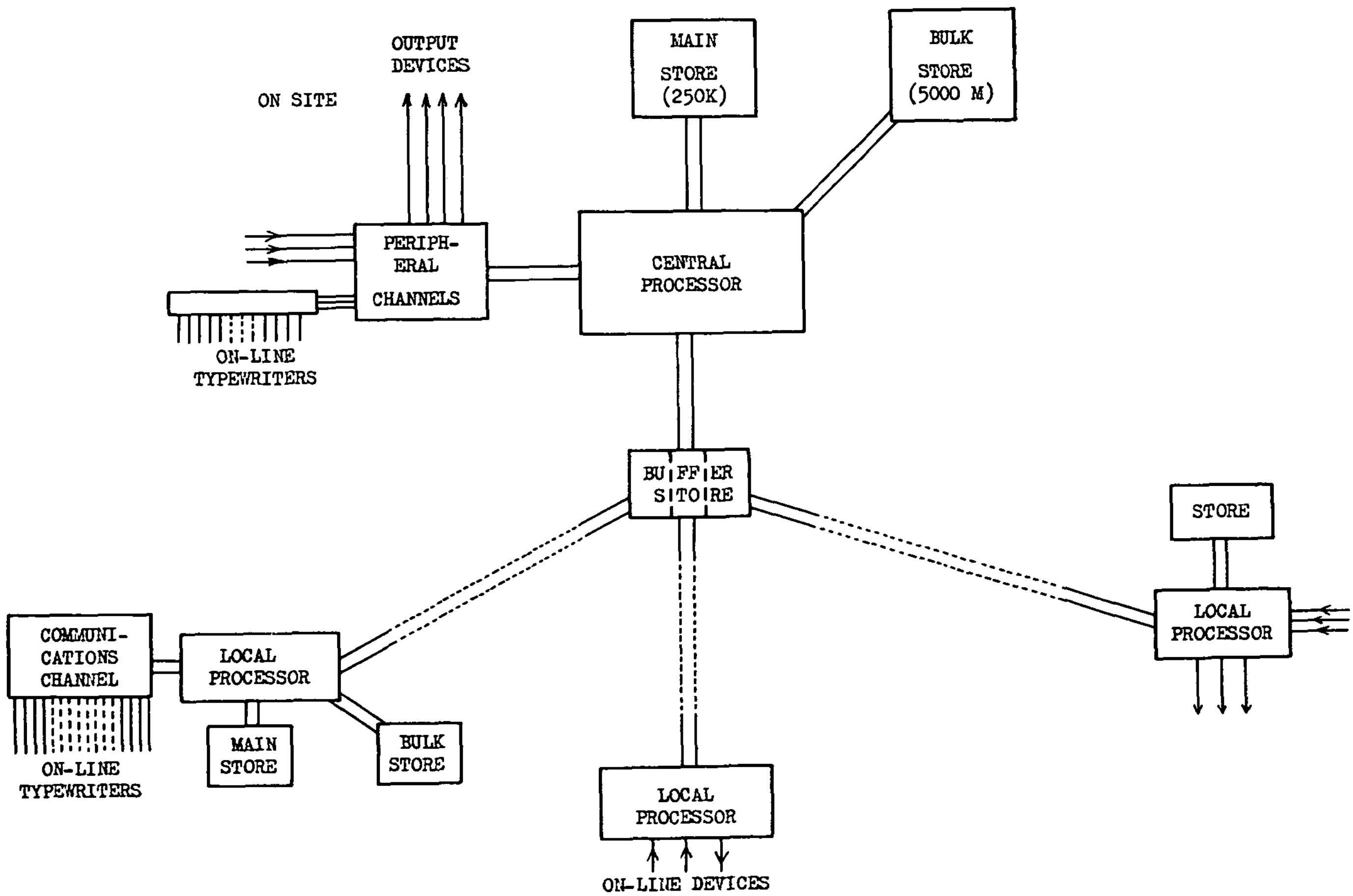

Fig. 5—Extended system

Man-computer systems

Leaving on one side for the moment the problem of how these many requirements are to be met, let me return to one of the earlier themes of my lecture, namely the need to study and develop efficient man-computer systems. An immediate necessity here is to improve our means of communicating with computers. I am not thinking solely of the languages used for programming. It is now appreciated very well that a computing system is far more than hardware, that the range and extent of the programming systems which go with it are equally important. As we are so well aware, a computer must not be considered to have been delivered to the customer until this software is in a proved working condition. Till then we may have a machine which works, but one akin to a two-year-old child to whom one can converse only in baby-talk—a potential adult, it is true—but needing a long and expensive period of education. Now we insist on having our computers delivered at least as teenagers. The universities of this country can take a good deal of credit for the way in which computers have grown up.

Programming languages need to be improved, but more important is the ability to interact with the computer during the execution phase of a program. This was common practice in early days when computers were comparatively slow, but as speed increased by factors of 10 or 100 so it became necessary, to take advantage of this speed, to limit interference to a minimum. The practice of closed-shop operation grew, and jobs are now commonly run in batches, often being preprocessed on a subsidiary machine before transfer on magnetic tape to the main computer. It is implicit in this that once the program is assembled in the computer, control can be handed over wholly to it. For the development of long, complicated programs this may be a protracted and tedious business.

An alternative approach is exemplified by the Atlas system, in which an elaborate supervisor program takes charge of the input, assembly, and execution of programs entering the machine simultaneously via a number of input channels, and tries to make the best use of the main store and the central processor. **Figs. 2** and **3** illustrate some aspects of the development which has taken place in five years.

With the arrival of time-sharing machines the situation has changed considerably, and we can begin to think anew about the advantages of closer co-operation. Experiments in this direction are proceeding in a number of places, universities and elsewhere, in the United States; in this country very little has yet been possible, mainly because of the lack of computers in universities with adequate fast-store and magnetic-tape facilities. One example of this development is project MAC at M.I.T., illustrated in **Fig. 4**.

If all the complex requirements of university work are to be handled successfully in the future it is almost axiomatic that it must be possible for a central computer or computer system to be used simultaneously by a large number of users, many of whom may be remotely situated. It is easy to devise such a system (as, for example, in **Fig. 5**), but a great deal of research has yet to go into the design of hardware, the logic of the system, and especially into the software which will control the flow of work and enable programs and data to be relocated at will within the system. I believe that with the present trend in computer development a system can be developed to meet very diverse requirements with a substantial degree of efficiency, reliability and accuracy. At least it should be clear to everyone that the computer revolution has hardly begun.

References

BRANDON, D. H. (1964). *Computers and Automation*, Vol. 13, No. 8, p. 22.

DE BRUIJN, W. K., FRIELINK, A. B., and SCHEEPMAKER, B. (1963). *Development of the Computer Market in Europe* (Netherlands Automatic Information Processing Research Centre).

FEIN, L. (1961). *American Scientist*, Vol. 49, pp. 149–68.

KEENAN, T. A. (1964). *Comm. A.C.M.*, Vol. 7, p. 206.

LICKLIDER, J. C. R. (1962). *Management and the Computer of the Future* (ed. Greenberger, M.I.T. Press and John Wiley), p. 208.

PERLIS, A. J. (1962). *Ibid.*, p. 210.