THE COMING OF AGE OF COMPUTER TECHNOLOGY

Inaugural Lecture of the
Second Professor of Electrical Engineering
delivered at the College
on October 21st, 1971

by

DAVID ASPINALL
M.Sc., Ph.D. (Manchester), C.Eng., F.B.C.S., M.I.E.E.

UNIVERSITY COLLEGE OF SWANSEA

Gomerian Press - Llandysul

UNIVERSITY COLLEGE OF SWANSEA

THE COMING OF AGE OF COMPUTER TECHNOLOGY

Inaugural Lecture of the
Second Professor of Electrical Engineering
delivered at the College
on October 21st, 1971

by

DAVID ASPINALL
M.Sc., Ph.D. (Manchester), C.Eng., F.B.C.S., M.I.E.E.

THE COMING OF AGE OF COMPUTER TECHNOLOGY

The last century has witnessed the birth of many technologies which have grown up through infancy and adolescence to reach maturity. The days of infancy are marked by the first halting steps taken by the early pioneers, men with the vision to conceive how a basic human desire or activity could be served by making an invention. This may be based on existing manufacturing expertise or require the development of a whole new technology which often extends beyond the bounds of the first invention. Such pioneering inovations are explorations which begin because of the inventor's intuition that there is some new truth to be found—not always because he believes that the discovery is of immediate value. The inventions are often the toys of the inventor, built because they can be built rather than because they are known to be useful. During the infancy of a technology many weird and wonderful devices are constructed and tested. Some of these reach the adolescent stage, when the most useful examples are tested by a widening range of people who are attracted, to some extent, by the novelty of the invention but also by the need to use it. As the adolescent stage progresses, the inventors are forced to devote more effort to develop the usefulness of their toys. The users are beginning to be selective in their choice of inventions and to appreciate the uses to which they can be put.

The inventions best fitted to serve the most important uses survive, and definitive forms for accepted devices become extant. At some time during adolescence, the collection of inventions reaches the status of a body of knowledge, which becomes the basis for design. Design criteria and principles of good practice begin to emerge. Design teams grow around the successful inventors, and the product of each team has its own characteristic style. There is still a tendency for inventions to be made for their own sake, but each new invention is now measured against performance and economic criteria before it is adopted. The motive for new developments becomes the need to make the technology more acceptable to the human

race ; to adapt the technology to the human being. At this point the technology can be said to have ' come of age '. Later, during maturity, the style of the early design teams is lost as generally accepted methods of design and construction become established. There is still development of the technology, but this goes on at a much slower pace. It depends upon the gathering of knowledge about the use to which the technology is put, the emergence of new uses for the technology, the discovery of new manufacturing techniques, and the ability of a new invention to provide extra benefits within the existing framework of design expertise, capital investment in manufacturing plant, and above all, the specification which the user has grown to demand.

An important characteristic of a mature complex technology is the proliferation of devices and systems which grow because of the existence of the original invention and which are necessary for its full exploitation. The railway engine requires a network of lines and associated signalling systems to provide a safe, reliable, rail service ; the modern motor car would be useless without a good road system ; and an airliner requires airport services, air traffic control, and air navigation systems to be effective.

The original invention at the heart of a technology can be thought of as its "hardware", whilst all the supporting services which enable the hardware to be fully exploited by man may be thought of as the "software". Maturity sees developments in both the hardware and the software. For successful growth, both must harmonise and develop together.

It must be remembered that without the existence of the hardware in the first place, the software would be impotent. Furthermore, a single development in the performance of the hardware is likely to produce a more significant impact on the power of a technology than developments in the software could ever achieve. (The replacement of the poston-engined airliner, such as the Dakota, by the turbo-prop airliner such as the Viscount, revolutionised air transport technology). Yet each hardware breakthrough requires a development of its supporting software before full benefit can be enjoyed. If the

softweare did not exist, the hardware would remain the inventor's toy.

This interplay and interdependance of hardware and software is perhaps most evident in one of man's more complex technologies, that of information processing so-called Computer Technology. Let us now trace its evolution through the early years until the definitive form of the basic hardware emerges, and then follow the development of electronic computers up to the present-day, at one major British centre. Along the way we shall attempt to identify the components which have given computers the potential power to make an impact on society and the concepts which unleash this power. We shall see how the technology has developed into a hierarchy of levels of knowledge and expertise as it approaches maturity. Finally, we shall discuss the role of a University in furthering its future development.

In the beginning, the abacus was invented to assist in computation. There was then a long pause in the pioneering phase until Napier, Leibnitz and Pascal invented aids to computation in the 17th century. The next major advance was made by Charles Babbage in the early 19th century. Charles Babbage, born in 1791, was elected to the Lucasian Chair of Mathematics at Cambridge in 1828, and held the post for eleven years without giving a single lecture in the University. In 1822 he had constructed his first "difference engine" which was intended to mechanise the computation required in the production of mathematical tables. He planned to build a much larger machine, working to twenty decimal places and sixth order differences. Government support was obtained for this project, but was withdrawn in 1842 with the work unfinished. However, a model of the "difference engine" was demonstrated at the International Exhibition of 1862—and a difference engine was later used for calculating the life tables needed for computing Insurance premiums.

There were many reasons why the ambitious "difference engine" was never completed. It required accurate machining of mechanical parts before the necessary techniques had been established. One of Charles Babbage's co-workers was

Whitworth who, after this salutory experience, went on to lay down the principles of mechanical engineering practice. The invention was ahead of the technology. If we remember that this work began before the era of the steam railway*, we can imagine the technological difficulties which Babbage encountered. Society was beginning to realise that physical labour could be assisted by machines, but it was not ready to consider how the drudgery of mental and clerical work could also be alleviated. Babbage realised that the difference engine was capable of assisting in the computations required for one, and only one, particular problem—the production of mathematical tables. The mechanism would have to be altered to solve a different class of problems. The difficulties of manufacturing that one piece of hardware, combined with the appalling prospect of starting all over again to manufacture hardware to solve a different problem, caused him to think again. It led him to suggest the idea of a basic computing mill, capable of performing a few standard operations on numbers held in a mechanical memory, and a mechanism whereby the mill could be controlled to carry out these operations in the sequence required to complete the computation. The numbers in the memory and the sequence—or program—of operations could be readily changed to make the mill solve many problems. The mechanism he used to effect the program control was the Jacquard punched roll system, used to control the patterns produced by weaving looms. Just as one Jacquard loom could be made to produce a wide range of patterns, so could the mill be made to complete a wide range of computations. This program-controlled mill he called the "analytical engine" which he invented in 1833.

The ideas about the exact form of the engine and the methods of programming it were developed in collaboration with the daughter of Lord Byron, Ada Augusta Countess of Lovelace, who was the first computer programmer.

Charles Babbage's enthusiasm for the analytical engine

*It is interesting to note that Babbage collaborated with I. K. Burnel of Great Western fame and designed the first dynamometer car.

deflected him from his ambitious difference engine, but the inadequacies of the technology let him down again, and he died one hundred years ago, in 1871, without seeing his engines in operation. But he left behind the concept of a general-purpose computer with programmed control, which is the basis of all modern computers.

The next pioneer was Hollerith who, in the 1890s, took the Jacquard punched roll and invented the punched card to assist in the computations required by the U.S. census. This idea was later developed, and found application in commercial data processing.

The advance of telephone technology led to the development of the electro-magnetic relay which was used by the Bell Telephone Laboratories in the construction of a computer in 1933. Relays were again used by Harvard University in the construction of a computer similar to the analytical engine. This was completed in 1944—one hundred and eleven years after Babbage conceived the idea.

Radio and the explosive growth of electronics during the second world war, produced the technology which made it possible for the Moore School of Electrical Engineering at the University of Pensylvannia to build the first electronic computer the ENIAC, in 1946.

By now, the form of the computer was beginning to emerge and was clarified by the contributions of Turing and Von Neuman. Turing put Babbage's concept on a sound theoretical basis by developing the idea of a universal logical computing machine which could be transformed into different particular machines by the superposition of programs. The hardware of the universal machine could be made to function to suit the particular requirement by the superposition of software. Each particular combination of hardware and software existing as a machine which can be more readily understood by its user, who is able to control it by writing simple programs in a language which he readily understands.

Von Neuman considered the structure of such machines. He realised that the most suitable number representing scheme within the hardware was the binary system. Data could be

efficiently represented as combinations of the Boolean truth values by the absence or presence of an electrical voltage or current. Logical, hence arithmetic, operations could be accomplished by the opening and closing of electrical switches. Furthermore, the instructions to control the sequence of the program could also be represented, or coded, in a similar way. Thus both types of information within the machine—data and instructions—could be represented by the binary system. The next step was to suggest that the memory unit attached to the mill of the engine be used to store both the data and instructions required to execute a particular task. This stored program computer offered many advantages ; the program roll could be read once into the memory and the program executed from the memory at a high rate. The program could be made to alter itself within the memory ; a repetitive operation could be programmed as a single sequence, obeyed many times by instructions which permit recycling of the sequence over and over again ; the ability to test the state of data in the memory and, as a result of the test, cause the next instruction to be taken out of sequence, i.e. to conditionally jump to another part of the program, made it possible to break out of a closed cycle and proceed to the next part of the program. These are the basic ingredients of the computer as we know it today.
Such a machine is shown in Fig. I. The data and instructions necessary for the computation are first read from the input punched roll into pre-assigned locations, or addresses, within the memory. To obey the program, the Program Control Device presents the address of the first instruction to the memory ; the instruction is read into a Present Instruction Register, from where its components are routed to present the address of the Data to the memory, to obtain the operand required by the MILL, and the MILL operation is selected. The operand is processed by the MILL and it is then ready for the Program Control Device to access the next instruction, and so on. During the course of the computation, answers are formed in the memory from whence they are written on to the Output Roll, as required.

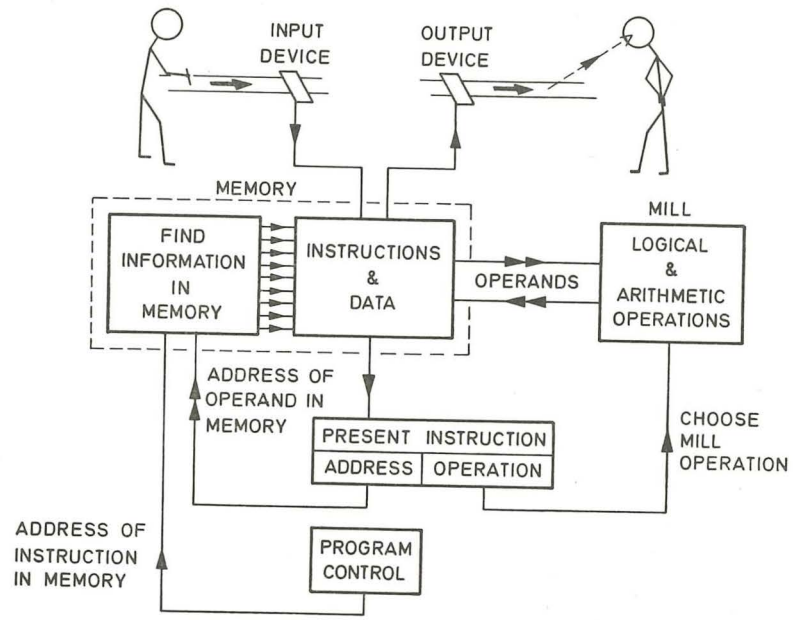

FIG. 1 BASIC STORED PROGRAM COMPUTER

This complicated diagram can be subdivided into three major sections :-

(1) Input and Output mechanisms for communicating information between the computer and its user.

(2) The Memory to store all the information required by the computer to complete its task.

(3) The Mill, and the circuits to route information within the computer and to control the various stages in the process.

To trace the development of the technology of each of these sections, let us consider the evolution of computers at the University of Manchester. This evolution is typical of that which occurred at other major centres in the world.

In 1947 Professor F. C. Williams returned to the Department of Electrical Engineering in the University of Manchester from T.R.E. with the idea that a memory could be based on the persistence—or afterglow—of a cathode ray tube (CRT). He demonstrated that a pattern of binary digits could be stored on the face of a CRT and the persistence could be maintained by reading each digit in turn, before it had died away completely, and then restoring it to its original state of brilliance. This tube formed the basis of the first experimental machine which obeyed its first program in the autumn of 1947. It had a simple mill—or accumulator—and arithmetic operations, routing, and related control were accomplished by thermionic valve circuits. The input mechanism was a simple binary keyboard, and the output was obtained by visual inspection of the CRT. Williams and Tom Kilburn (now Professor of Computer Science at the University of Manchester) went on to construct several prototype machines until a large-scale machine was built in 1949—a development which formed the basis for the Ferranti Mk. I Computer in 1951, the first commercially available computer in the world. This machine was based on the CRT memory which, though well-matched to the speed of the accumulator circuits, was of limited size, storing only 512 numbers of data or instructions. The size of the memory was increased by adding magnetic drums on which 650,000 binary digits, or bits, of information were stored in much the same way as speech is recorded on a magnetic tape. Input facilities were provided by a photoelectric paper tape reader, whilst output was obtained by a mechanical paper tape punch and teleprinter. Use of the prototype machines had created a demand for certain facilities to improve programming efficiency the most notable of these being the address modification, or indexing mechanism*. Developments in thermionic valve

*It is interesting at this point to record the contributions made by three Welshmen. D. B. G. Edwards, of Pontypridd (now Professor of Computer Engineering at the University of Manchester, who has contributed in many ways to the development of computing at Manchester) ; G. E. Thomas, from Port Talbot (now Director of the Edinburgh Regional Computer Centre) ; and E. T. Warburton, also of Pontypridd (now doyen of computer designers in British Industry). All three were involved in the design of these early machines, which owed much to their originality, determination, and engineering skill.

technology made it possible to produce a faster, more reliable, computer which was developed by Ferranti Ltd. as the Mercury computer in 1956. This machine also contained an improved mill capable of performing operations on numbers represented in floating-point from—a further demonstration of the users' influence on the machine specification.

An experimental transistorised computer was operating in 1953 which influenced the design of such machines by British industry, but the next major advance began in 1958 at the start of what became the ATLAS Computer.

By 1958 there had been ten years experience in the use of computers at Manchester. Two types of user were beginning to emerge. The first had large problems, which could sensibly use much of the machine time based on a few programs which, once written and tested, would run many times on different sets of input data. The second had a list of many small problems each requiring a different program, and probably only one run on the computer to produce results. The large problem users were prepared to take a long time to understand fully the intericacies of the computer, to acquire the necessary programming skills, and to learn the language of the hardware. The small problem users found this less rewarding. They demanded that the hardware be augmented by the superpositioning of software to present to them a language which they could quickly learn to use with accuracy. These so-called high-level languages, or Autocodes, are translated into the hardware language by Compiler programs in the machine. The problem facing the ATLAS designers was to produce a machine which would provide facilities to assist the compilers in producing hardware language programs which would run efficiently when compared with programs written directly in the hardware language by skilled programmers. These facilities had to be provided in a way which would not cheat the hardware language programmer by reducing the efficiency of the machine.

The Von Neuman type memory by this time was provided by a high-speed ferrite core memory, in which each bit of information is remembered by the direction of magnetisation

of a ferrite ring, backed by a large magnetic drum memory. A similar memory hierarchy existed on the earlier machines in which the programmer had to arrange for information to be brought down from the drum (it was actually located on the floor above), to be processed in the high-speed memory. Much skill was required of the programmer to anticipate the time at which the relevant piece of information passed under the fixed reading station, located above the surface of the rotating drum ; also the working space of the memory was restricted to the size of the high-speed memory. This led to many programming problems connected with the movement of information up from the high-speed memory to make room for that brought down drom the drum. The problem was solved by allowing the ATLAS programmer to think of the memory working area as one level equal to that available on the drum. In practice, the computer Mill still operated on information contained in the high-speed ferrite core memory, and transfers between the drum and the core were effected automatically on demand. This one-level memory concept was a major advance which had unforeseen ramifications—which we shall consider later.

The successful improvement in the ease of computer use was bound to increase the number of users and jobs to be processed. Each job has to be fed into the computer and produces data which must be fed out. The traffic through the peripheral input/output devices would increase. Whilst there had been many improvements in the efficiency of existing peripherals, and new ones, such as magnetic tape, line printer, and graphic displays were being rigorously developed, there was still a large discrepancy between the speed of any one peripheral and the computing rate within the machine. The only way to keep pace was to attach many peripherals onto a single computer and allow simultaneous transfers between them and its memory. This Time Sharing of a powerful computer between many concurrent activities was made possible by the inherently high speed of the computer plus hardware facilities which permitted special software—known as Supervisor or Operating System—to be superimposed to manage the flow of information.

These were some of the problems faced by the ATLAS project. They would not have been solved but for the significant advance which had occurred in the technology of computer hardware. We have already noted the development of the ferrite core memory. This offered greater reliability and higher speed than could be obtained by the CRT. The other major development was that of semiconductor devices, the diode and the transistor, which supplanted thermionic devices as the basic components of the circuits in the Mill, routing and control sections of the computer. These were more reliable, consumed less power than their predecessors, and offered the possibility of parallel processing of information within the computer. Numbers could be added, not one digit at a time in a single circuit, but all digits at once by many circuits. The smaller size and inherent speed of these devices caused the designers to be concerned about the path length of information routes and the time taken for an electrical signal to pass along a piece of wire, determined ultimately by the speed of light.

During the infancy of this project, many semiconductor circuits were invented and evaluated to produce the range of basic circuits which could be constructed in sufficient numbers to build the machine. Different methods of using the ferrite core were tried until the most suitable memory was designed. Designers of the Mill and related circuits experimented with different ways to use the basic devices until a design philosophy was established. A prototype machine was built to test the major technological innovations.*

The next phase was marked by the drawing up of the final design specification, followed by the long process of design documentation, construction, inspection, and commissioning of

*This is one of the most exciting phases of a major project, similar in its emotive significance to when your first child beigns to walk, or learns to read. When he is taking his first steps you wonder if hw will ever wakl, until at last one joyful day dawns when he walks as if he had been doing it all his life. After this day you quickly forget that there was a time when he could not walk, and it is taken for granted. So, in the development of a computer, there is a time when it cannot add two and two together. One feels it should be able to, but it cannot. One makes many tests fo find the reason, until the last wire finally falls into place and, amid great jubilation, it adds correctly. From then on one takes it for granted, and goes on to face many other tribulations and joys.

the machine to reach a stage of reliability at which it could take on its computing load. During this phase, principles of sound engineering practice had to be established, and the existing computer used to assist in the production of design documents and to simulate some of the more complicated designs. Maintenance engineers had to be trained and suitable test procedures evaluated so that the computer could continue to give reliable service. The stage of maturity was reached during the early 1960s when the software, comprising both compilers and supervisor, was able to give the users the facilities they required ; and the hardware level of reliability was such that the shift maintenance engineers were able to clear faults without calling the designer out of his bed in the wee small hours of the morning.

The first production model of the ATLAS provided computing facilities to the University until it was finally switched off last month. Several production models were manufactured and are still in use in important computer centres in the United Kingdom. Some of the concepts pioneered in ATLAS were ahead of their time and have only recently been incorporated into other commercial computers. It became the paragon with which all the modern large computers are compared.

Once the computer was providing a service to users, the engineers were naturally banned from trying to improve its performance by testing new circuits or devices since during, and shortly after, each test the impaired reliability of the computer would be intolerable, and such changes to the hardware would have serious repercussions on the software, and hence the user programs. The software writers were similarly banned from making changes to proven software, but were able to add features and monitor the efficiency of the total system and the way in which it operated, to give guidance in the drawing up of the specification of its successor.

By 1966 it became possible to review the ATLAS design and its usage in the light of developments elsewhere in Computer Technology. Most of the computer users never saw its hardware. They tossed their programs into a wire tray on the floor below and returned a few hours later to collect their output

from the same wire tray. Some users never came to Manchester their local terminal equipment being linked by telephone line direct into the computer. The remoteness of the user presents difficulties during the program development phase when programming faults are detected by the machine and reported to the programmer. The delay between handing in the program and obtaining the report is often inconvenient and distorts the programmer's thought process. Other computer centres provided interactive program development facilities by which the programmer could sit at a teleprinter console and key his program direct into the machine, check it, and make the necessary corrections until it was satisfactory. Fully to incorporate these facilities on top of the existing batch processing service required more extensive resources than were available in the ATLAS System.

The hardware of a large time-sharing computer is shown in Fig. 2. It consists of a MILL and its associated circuits now known as a Central Processing Unit (CPU), a main memory to hold the information required by the CPU during its execution of the program, a large backing memory holding data and software, the peripherals for the batch processing service, card and tape readers and punches, line printers, etc., and the interactive consoles—some local, some remote at the end of telephone lines, with a computer (Peripheral Processing Unit) to assist in the management of the information flow. If one stood amidst the peripheral equipment one would see tapes being fed-in via several tape readers, at the same time as cards are being punched with output data, at the same time as characters are being printed, at the same time as puctures are being painted on the graphics terminals, and at the same time as several programmers are developing programs at their interactive consoles. This hive of activity depends on the inherent high speed of the CPU hardware, plus the ability of the operating system software to marshall and co-ordinate.

What the operating system must do is to transform the hardware of Fig. 2 into the conceptual system shown in Fig. 3 in which each of the activities or processes is granted a virtual

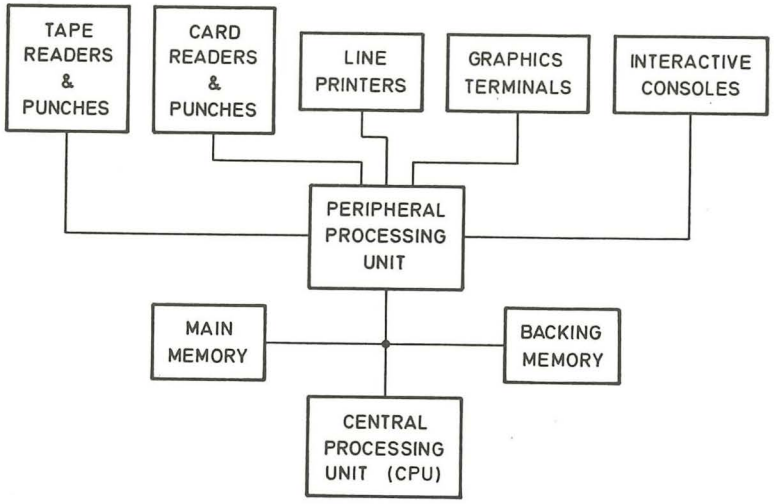

FIG. 2 BLOCK DIAGRAM OF TIME SHARING COMPUTER - ACTUAL

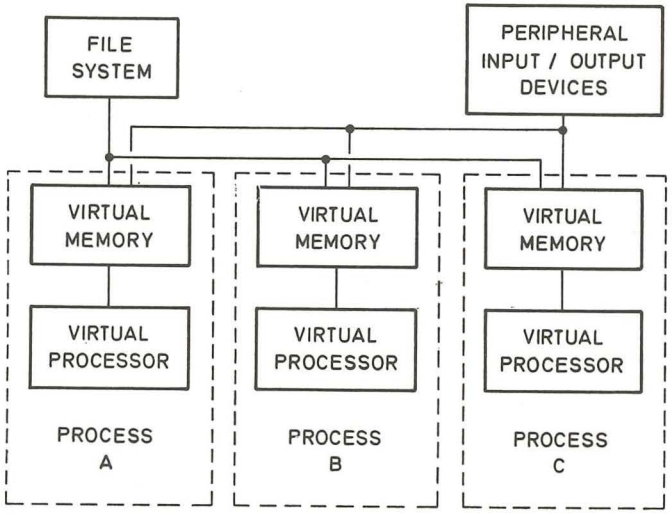

FIG. 3 BLOCK DIAGRAM OF TIME SHARING COMPUTER - VIRTUAL

memory and a virtual processor, which have access to a common file store and peripheral devices.

Let us consider three typical concurrent processes :

Process A : A large program requiring much CPU time.
Process B : A program to control the flow of data to a line printer.
Process C : A program under development at an interactive console.

Let us assume that the CPU and memory are operating on Process A which is in the middle of a matrix inversion (say) and that the line printer becomes ready to receive new data. Process A will be suspended and Process B entered to copy data from the memory to the line printer. When Process B has finished its task, Process A will be re-established and allowed to continue with its matrix inversion where it left off. After a while, Process A may be suspended to allow Process C to test the present status of the program under development, after which there is a return to Process A, and so on. In spite of all this concurrent activity, the line printer is able to print lines continously ; the programmer appears to have the whole of a machine at his desposal to develop his program, and whilst Process A may take longer to completion than if it were the only process in the machine, even the user of large programs benefits since, whilst Process A is being executed he may also be getting output from another of his programs by Process B, and the total turn-round time, from putting the program in, to getting the answers out, will not suffer appreciably.

In order for the operating system to effect the mapping of Fig. 3 on to Fig. 2, the hardware must provide facilities which permit the rapid suspension and re-establishment of a process, to allow information in different virtual memories to exist side by side in the same actual main memory, and to prevent one virtual processor from accessing private information in the virtual memory of another, yet allow processes to share common information. A key to the solution of these complex problems is the One Level Memory concept developed on ATLAS to disguise its main memory-backing store hierarchy from the

programmer. This concept enables the actual processor to offer, to all concurrent processes, a large memory space which may be sub-divided to give each process its virtual memory. Overlapping of the virtual memory of one process with that is of another allowed to permit sharing of information, and protection facilities can be readily incorporated.

Whereas any such concept can be implemented by superposition of software on to a machine, the full benefit can be enjoyed only if special facilities are built into its hardware. For this to happen, the hardware designer must be aware of the total software problem, attempt to isolate those stages in its solution which can benefit from the provision of special facilities, and have a complete grasp of the potential power of the hardware technology and its ability to provide them efficiently.

In the case of the One Level Memory concept, a major problem is to translate the large address of the virtual memory into the smaller address of the actual memory. The hardware solution is to provide a special address translation mechanism known as an associative or content addressable memory. Such a mechanism became an economic possibility after the development of integrated semiconductor circuits. These superseded the discrete semiconductor circuits of the ATLAS and offered more complexity at less cost, with higher reliability at higher speed.

The features of the new circuits may be exploited in many styles. At one extreme the hardware designer may exploit the speed and complexity to produce extremely fast arithmetic and data processing facilities with little regard to the needs of the software. This results in a very powerful CPU capable of a high rate of computation which can be exploited by the writer of programs in a low-level language, but which deny the high-level languages all but a fraction of this rate. At the other extreme, the total software requirements may impose a design specification on the hardware designer and he must endeavour to make the best use of his technology to meet this specification. The most successful style, however, is to allow the hardware and software considerations to interact to produce a specification

which satisfies both ; to obtain a machine structure in which the special facilities may be provided by either software or hardware ; by software in the small, low cost, machines, and by hardware in the larger, more expensive, more powerful, machines towards the top of a manufacturer's range.

By 1966 such a specification was beginning to evolve at Manchester. Methods of using and interconnecting the new integrated circuits had become established, and memory technology had developed to keep pace by the introduction of plated wire memories to replace the high speed core memory, also cheaper core memories to provide low cost mass memory.

The University was again ready to embark on a major computer project and, with S.R.C. support plus the co-operation of ICL, the MU-5 computer is being constructed. It will soon be an operating system—more powerful than the ATLAS.

If we now consider the body of knowledge which makes up Computer Technology, we find a hierarchy of levels as shown in Fig. 4. At the lowest level—the circuit level—is the basic electrical engineering of the components of the hardware. This involves a knowledge of the desired characteristics of the circuits which perform logical operations and provide the various levels of memory ; how these characteristics can be provided by available devices and the way in which such devices need to be developed to obtain improved performance ; how to solve the electrical problems which arise when such devices are connected together, plus the technology of the peripheral equipment. The next level above, logical design, includes a methodology for connecting the basic circuits together to perform the various operations in a computer such as the addition circuit of the MILL and its control circuits. Above this is the machine structure, or architecture level. This is where the interaction between the hardware and software takes place ; where the requirements of the user and software writers are considered to arrive at the specification of the logical units which will best make up the total hardware of the machine. For this level to function properly there must be a clear understanding of the software problems and the ability

to isolate those which can be effectively solved within the capabilities of the hardware. Next comes the system software, the writing of the compilers, and the operating system and other programs which enable the last level—the user programmers—to make efficient use of the machines.

The performance at each level is dependant on the resources which can be provided by the next lower level. Ultimately, the performance of the user programs depend upon the power of the circuits produced by the lowest levels. The way in which the performace of these circuits has developed over the years is shown by the graph in Fig. 5. The vertical axis measures the rate at which an addition can be performed on numbers of ten decimal digits. Each point plotted shows the rate obtained by the addition circuit of a Manchester computer in the year when the hardware was first demonstrated. In 1947 the first adder could operate at the rate of one thousand per second. By 1952 improvements in thermionic valve technology and design skill resulted in a jump to thirty thousand per second. The next jump to three million per second was achieved by the exploitation of semi conductor devices, and the present rate of twenty million per second is made possible by semi conductor integrated circuits and improved constructional techniques. Each of these advances was matched by a corresponding advance in the technology of the main memory and the development of strategies to ensure that the flow of information between it and the central processor unit kept pace. It is interesting to note that rate of growth prior to 1958 was greater than that obtained since then, suggesting that we are approaching the speed limit of electronic technology, which is when we can no longer reduce the path length between circuits, and must accept the ultimate limitation imposed by the speed of light. A modern integrated circuit delays the passage of information through it by some 2 nanoseconds (2.10^{-9} secs). This is similar to the delay along a piece of wire one foot in length. The physical size of the circuit and the distance between circuits begin to have a serious effect upon the speed of the machine. The size is determined to some extent by the fact that the circuit has to be handled at various stages of manu-

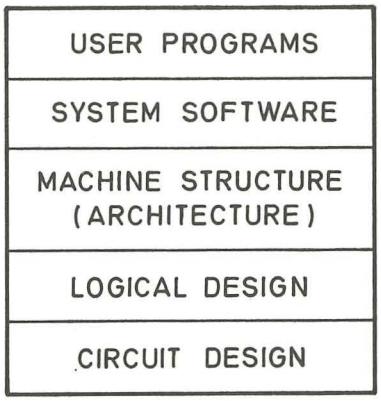

FIG. 4 HIERARCHY OF COMPUTER TECHNOLOGY

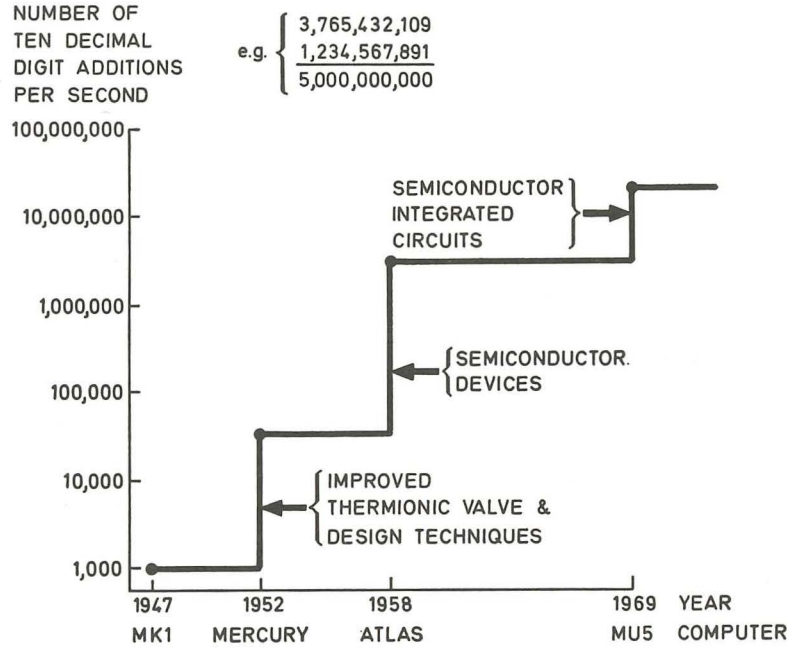

FIG. 5 GROWTH OF COMPUTING POWER

facture and test ; it must be in a package large enough to be picked up by average sized fingers and thumbs. The distance between circuits depends upon the size of the circuits and the space to connect the interconnecting wires, power supply, and allow cooling fluid to take the heat away. Any further reduction in circuit delay will have to be associated with the ability to pack more circuits into the package, between finger and thumb, if possible requiring less power and cooling than present packages. Further speed improvements will be obtained by developments of machine structures which permit a proliferation of parallel processing circuits to be exploited by the system software.

One might ask why there is this thirst for speed. There are two answers to this question. First, there are important large problems which demand it, and secondly, it is cheaper to serve several small users on one fast machine than to provide each with his own slow one. The thrust towards a fast large computer leaves many developments in its wake which benefit the machines required in situations which the large, fast, time-shared machines cannot reach-process control computers, vehicular control computers, dedicated laboratory instrumentation computers, etc.

The growth in speed since 1947 has been the greatest single factor which has given the upper levels of the computer technology hierarchy the nesessary power to produce the present impact of computers.

If one examines the present state of development of computer technology, one finds the mature levels at the bottom of the hierarchy, maturity decreasing as one ascends.

The circuit level is well aware of the needs of its immediate user, the logic design level. It is also well aware of the constraints placed upon it by the known laws of physics and has the advantage of many yearx of engineering experience in the application of electronics technology in the fields of communication, Radar and control. Design standards and codes of good practice which were established in these fields have formed the basis for sound computer engineering. This level awaits the discovery of a new physical penomenon, with an information

processing capability, before it can return to a pioneering phase.

Logical design is restricted by the logic circuits which can be engineered economically and reliably by the circuit level. The laws of logical design based on Boolean algebra are well understood and principles of good design are well established, now that a logical design can be fully tested by means of simulation before the circuit is built. The functional circuits which make up a computing system are well understood and can be specified to the satisfaction of the logic designers and the computer architects. Whilst there is always scope for development at this level resulting from its interaction with its two neighbours, it awaits the birth of a new idea—possibly based on adaptive logic—for it to return to the pioneering phase.

The machine structure level is taking longer to reach maturity. This is due on the one hand to the flexibility offered by the functional units produced at the logic level, and on the other to the complexity and flexibility offered to the system software by any computer system. Furthermore it takes several years for a computer system to be conceived, designed, constructed, and tested in the user environment ; and the environment itself changes as the user requirements develop.

It cannot reach maturity until the next level above is able to define the computer structures which it requires. This level, system software, is still developing. It has to keep pace with a shifting user requirement and new ideas resulting from its interaction with the computer structure level. It does not have the benefit of sound engineering practice, partly because most of the workers in this field come from a non-engineering background and enjoy experimenting with their new toy. There are signs that principles of software engineering are becoming established as the innovators realise that their ideas have to be implemented by a body of people organised to build, test, and maintain the software in the same way as any other piece of engineering. When the software functional units can be specified as precisely as the hardware functional units, then the software and machine structure levels will have reached maturity.

Though these principles of sound engineering practice will be established and a semblance of maturity will be achieved, these two levels still have to interact with the top level, the user. There will always be pioneer users since the potential field of application of computer technology is not restricted by the nature of the technology itself but by what the human mind chooses for it. Man will continually discover new uses for information processing technology ; uses which we cannot perceive at present ; uses which are bound to involve developments in the upper levels of the hierarchy ; uses which will place new demands on all its levels.

A university which offers a course of instruction in Computer Technology must ensure that it can teach all levels of the hierarchy in a co-ordinated manner. In its research it can concentrate on a particular level or take a microcosm of the hierarchy by considering an area of application which involves all levels. Such a microcosm could be the application of computers in instrumentation. One example, in the medical field, is to provide computing facilities for the interpretation of chart recordings such as electro-encephlographs. We can all appreciate that much of the information on such charts is meaningless and that skill is needed to extract that which is important for the diagnosis. The doctor requires a computer, in the path of the signals between that patient and the chart recorder, to eliminate the unwanted information. This computer could be a general purpose computer with special circuits to translate the signals from the patient into the number representing system of the hardware of the computer. A supervisor program within the computer could manage the flow of information from several monitor points on the patient to the various processes which carry out the information reduction. These processes themselves could be written in a suitable high level language which provides the doctor with convenient, easy to use, control facilities. Such computer systems have been in use for some time, and experience has shown that a special class of general purpose computers is required to satisfy the many needs at an economic price. It is possible to write down the specification of such a machine and plan a

development project involving all levels of the hierarchy. Such projects are undertaken by industry, but certain ones can be fruitfully carried out by a University as a vehicle for maintaining competence and developing new ideas. The new ideas may occur at only one level of the hierarchy, but it is to be expected that they will result from a better understanding of the interaction of the levels. To begin with, these projects will involve the writing of special software for an existing computer, plus the provision of special perepherals. The next stage will be to design special computers to suit the particular requirements. This will become more economical as powerful, complex, integrated circuits proliferate. We may evolve away from the concepts of Babbage, Turing, and Von Neuman, and solve each new user problem by designing a special new machine from the lower levels up, providing only sufficeint hardware and software to satisfy each user's needs.

This approach will satisfy only a section of the computer technology users. Most will carry out their information processing on general purpose computers. In the past we have seen a range of general purpose computers to serve the many and varied user needs. However, we are beginning to appreciate that the economics of the situation dictate that user needs be served by a few large, powerful, computers and a proliferation of many small low cost machines. This polarisation of computers into large and small will come about partly because of the development of integrated circuits which will make it possible to place a low cost computer in a small package. (We are already seeing this impact in the field of desk-top calculators which are now based on integrated circuits and may soon be as commonplace as a students' slide rule), and partly because of the development of a data communications network which will make it convenient for users with large problems to have them processed on a remote large computer. Such a network will open the door to many new information processing developments. The information terminal may become as familiar as the telephone. The housewife could use the terminal to order her groceries, budget her account, decide on the day's menu, and control her cooker.

The future offers much scope for research at all levels of the hierarchy.

In conclusion let us attempt to sum up the situation. The technology has reached a stage at which one computer can perform as much arithmetic in one second as one man can be expected to perform in his entire working life. Furthermor, the computer is not expected to make an error in that one second, whereas there must be few men who have not made an error in their working lives. We are beginning to realise that there are facets to man's information processing needs other than sheer accurate arithmetic computation. Some of these needs are already being served to some extent in areas such as education, information retrieval, medicine, and commerce. As these needs develop, the technology will produce systems with a few simple controls which are easy to use. The basic components, the hardware and the software, will develop and combine to achieve this aim, and be able to cope with the new uses which man invents for the technology ; uses which will in the long term improve the quality of life.

Finally, I cannot think of a better way to end this lecture than to quote Professor F. C. Williams (who was this year admitted to an honorary degree by the University of Wales) :-

> "Lest the great achievement to date should make us swollen-headed, or alternatively, make us despair of further improvement, let me draw your attention to the small object glued to the centre of this card (a seed). This object is so small that I doubt if you all can see it, yet it contains all the information, data, and instructions necessary to control the manufacture of an unending and ever-increasing supply of roses."