

ATLAS

a new concept in
large computer design *by Dr. S. Gill.*

ATLAS is being designed jointly by Ferranti Ltd. and a team at the University of Manchester headed by Professor T. Kilburn. The first ATLAS, to be known as MUSE (Kilburn 1959), is scheduled for completion at the University by the end of 1961 from parts manufactured by Ferranti.

ATLAS will have the highest computing speed obtainable at a reasonable cost. The purpose of this speed is firstly to execute very large computations as efficiently as possible, and secondly to enable the control of peripheral units to be simplified by allowing the computer to attend to their detailed operation on an interrupt basis. The result is a flexible and economical system equally applicable to large scientific calculations and to commercial data processing. The design also incorporates an entirely new and sweeping approach to the problem of store utilization.

The project is made possible by technical advances made at Manchester University, including a fast-carry adder (Kilburn, Edwards & Aspinall 1959) and a rapid access fixed store (Kilburn & Grimsdale 1960). The normal computing store will be a form of ferrite core store employing two cores per bit (Edwards, Lanigan & Kilburn 1960). Each section of 4096 words will operate independently with a cycle time of 2 microseconds; thus, by using several sections, a mean access rate of over a megacycle is achieved (consecutive addresses appear in different sections). The average time to extract and execute a complete, single-address, floating-point addition instruction is expected to be 1.1 microseconds, but the actual time will vary depending on the amount of possible overlap with adjacent operations. The effective speed of multiplication is such that, for example, summing a polynomial will take from 5 to 7 microseconds per term.

In the fixed store, information is represented by the presence or absence of ferrite slugs in a woven wire mesh,

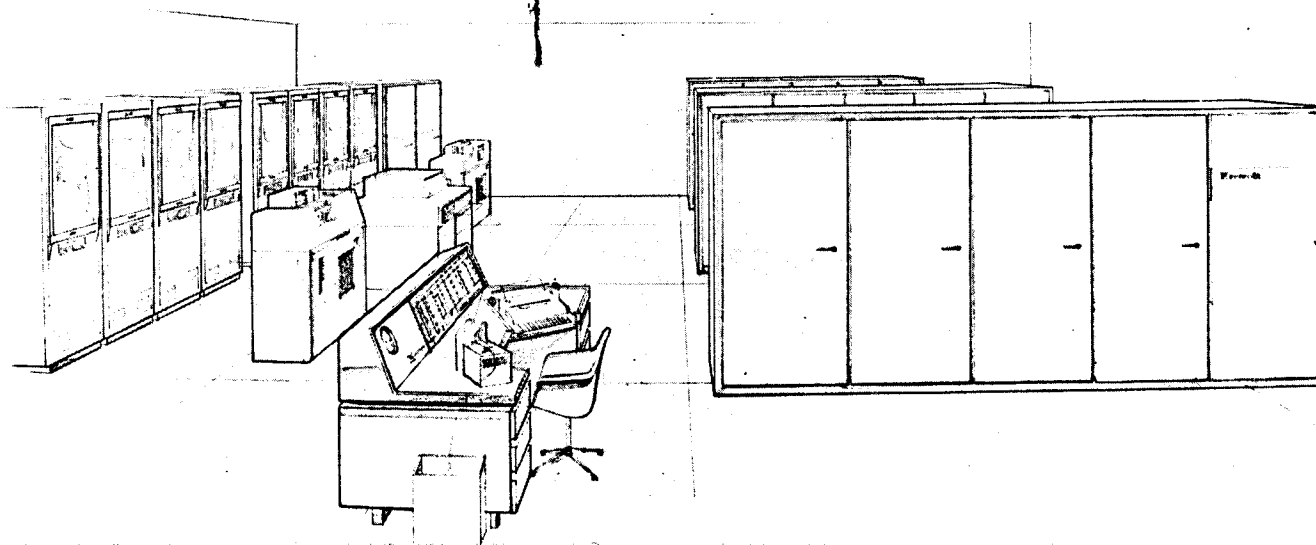
and the cycle time is approximately $1/5$ microsecond. This high speed enables many simple subroutines to be executed in times not much greater than those for some of the basic instructions, and the design is such that these subroutines are in every sense an extension of the basic instruction code. One bit in the function of every instruction indicates whether the instruction is basic or "Extracode" (i.e., to be interpreted by a fixed store routine). In the latter case the rest of the function part determines the entry point to the fixed store routine and the address part is available for use as a parameter. The format is thus exactly the same as that of a basic instruction.

The fixed store also contains routines for initiating and controlling peripheral transfers, for monitoring programs, for carrying out simple routine engineering tests, and for operating time-sharing between programs.

The word length is 48 bits. Numbers are represented in floating octal, with 8 bits allocated to the exponent. An instruction occupies a whole word, with 10 bits for the function, 7 bits for each of two 'B' or index addresses (arithmetic instructions can be double-indexed), and 24 bits for a single main store address. This generous allowance of instruction bits caters for the many developments of the system that are likely to occur in the future. Thus, there are 512 distinct functions that can be interpreted by Extracode routines; at present only about 200 of these have been reserved.

The indices are 24 bits long and are held in a special small core store which has a cycle time of approximately $\frac{1}{2}$ microsecond. There are nominally 128 indices but some of these are reserved for special purposes, e.g. to act as control registers and as the floating point accumulator exponent.

Of the 24 bits in a store address, the first is used to distinguish the users' store from the "private" part of the



machine (i.e., the fixed store together with a special section of core store which it uses as working space, and also certain registers associated with peripheral equipment, manual controls, etc.). The last 3 bits of an address are reserved for indicating the position of a 6-bit group or character within its 48-bit word. The remaining address digits suffice to cover approximately one million words in the users' store.

The most revolutionary feature of the machine concerns the manner in which the address digits are interpreted (after being modified by the addition of indices if appropriate). The store is regarded as being composed of blocks of 512 words each. Thus, 9 of the address bits determine the position of a word within its block, leaving 11 bits to identify the block. However, these 11 bits do not define directly the physical position of the block. Instead, each block is identified by an 11-bit floating label which is independent of its actual position.

With each "page" or block position in the core store there is associated a "page address register" to hold the label of the block which happens to occupy that page at any moment. Before every store access (whether for instruction or operand) all of these registers are consulted simultaneously and very rapidly. The register whose contents agree with the required block number responds to this enquiry, and its page is then used in the transfer. The consultation of the page address registers is done by special circuitry and does not limit the rate of transfers to and from the store.

Effectively this is a hardware version of the symbolic address idea, applied to complete blocks. However, looked at this way it is rather pointless since the labels are short and the blocks are larger than most items of data. Its real purpose is to provide a simple basis for the complete automation of the allocation of store space to several programs simultaneously and of the integration of backing storage (e.g., of magnetic drums) into the system. In this sphere, used in conjunction with the fixed store of ATLAS, it provides a simple and effective answer to most of the practical difficulties associated with time sharing and two-level store allocation in a large computer.

The way the system works is as follows. If a program has priority, so long as its data and instructions are in the core store, it may proceed without hindrance. If, however, it makes reference to a block which is not currently in the core store, none of the page address registers signals agreement, and a top priority interruption occurs, leading to a special routine in the fixed store. This routine has access to the page address registers and keeps directories of all blocks both in the core store and in the backing store (if any) and of the activities of all peripheral units. It is therefore in a position to administer the whole process of store allocation.

It will, for example, fence off blocks that are involved in magnetic tape transfers by putting dummy labels in their page address registers. (Indeed, these blocks will be identified for the magnetic tape controls by these dummy labels.) It will initiate transfers from the backing store when necessary, so that a program or its data may be held

partly in the backing store without any reference to this in the program itself. It will keep separate records of the space used by different programs and will ensure that each block is made available only to the program to which it belongs. (This involves, e.g., changing the contents of the page address registers when the computer is switched from one program to another.)

The only store protection which needs to be provided by hardware is that which is necessary to prevent the fixed store routine itself from being disturbed by an error in any of the programs that might be in the machine. This protection takes the form of a simple prohibition of references to the "private" part of the machine by all programs except those in the fixed store.

Among the effects of this method of organization are:

(1) There is no need to subdivide store addresses between programs which are to be operated simultaneously. Each program "sees" only its own addresses, which may range over the entire set of store addresses (although there is a limit to the total number of blocks which may be in use at one time).

(2) If it is considered economic to use a backing store (and it must be admitted that a million words of core store are not cheap), then an efficient utilization of the two-level combination is obtained without any attention from the programmer. Moreover, if the size of the core store is increased, the same programs are still valid and can immediately be run at greater speed.

Fixed store routines also allocate peripheral units to programs, and the usual interruption procedure is employed to permit these routines to attend to peripheral equipment when necessary. Magnetic tape and magnetic drum transfers of complete blocks are autonomous (using core store cycles as required) except for the initial preparation and final adjustment of block labels. Other peripheral devices are attached in a way which keeps the circuitry as simple as possible, the features of the transfers being determined mainly by fixed store routines. The system is therefore extremely flexible.

Currently the design envisages: a core store of 16,384 words as a minimum, which may be increased in units of 4096 words; up to 16 magnetic drums each holding 24,576 words; up to 32 magnetic tape decks communicating via 8 channels, each handling 90,000 characters per second; up to 8 line printers operating at 600 lines per minute; together with card readers and punches, punched tape readers and punches, xerographic printer and graphical display.

REFERENCES

- KILBURN, T., EDWARDS, D. B. G., AND ASPINALL, D., Parallel addition in digital computers: A new 'Fast-Carry' circuit. *Proc. I.E.E.* Pt. B, 106 (1959), 464.
- KILBURN, T., AND GRIMSDALE, R. L., A digital computer store with very short read time. *Proc. I.E.E.* Pt. B, 107 (to be published).
- EDWARDS, D. B. G., LANIGAN, M. J., AND KILBURN, T., Ferrite magnetic core memory systems with rapid cycle times. *Proc. I.E.E.* Pt. B, 107 (to be published).
- KILBURN, T., "MUSE". *Symposium on Large Machines*, Proceedings of the Unesco International Conference on Information Processing, June 1959. Oldenbourg Press, Munich.