

An American Federation of Information Processing Societies Publication

## THE ATLAS SUPERVISOR

*T. Kilburn and R. B. Payne*  
*The University of Manchester, Manchester, England*

*D. J. Howarth*  
*Ferranti Limited, London, England*

Reprinted From — **COMPUTERS - KEY TO TOTAL  
SYSTEMS CONTROL**

**F**erranti

# THE ATLAS SUPERVISOR

*T. Kilburn and R. B. Payne*  
*The University of Manchester, Manchester, England*

*D. J. Howarth*  
*Ferranti Limited, London, England*

## 1. INTRODUCTION

This paper gives a brief description of work originating in the Computer Group at Manchester University. Atlas\* is the name given to a large computing system which can include a variety of peripheral equipments, and an extensive store. All the activities of the system are controlled by a program called the supervisor. Several types of store are used, and the addressing system enables a virtually unlimited amount of each to be included. The primary store consists of magnetic cores with a cycle time of under two microseconds, which is effectively reduced by multiple selection mechanisms. The core store is divided into 512 word "pages"; this is also the size of the fixed blocks on drums and magnetic tapes. The core store and drum store are addressed identically, and drum transfers are performed automatically as described in Section 3. There is a fixed store which consists of a wire mesh into which ferrite slugs are inserted; it has a fast read-out time, and is used to hold common routines including routines of the supervisor. A subsidiary core store is used as working space for the supervisor. The V-store is a collective name given to various flip-flops throughout the computer, which can be read, set, and re-set by reading from or writing to particular store addresses.

\*A paper has been written on Atlas and it is hoped will be published by the Institute of Electrical Engineers.

The accumulator performs floating point arithmetic on 48-bit numbers, of which 8 bits are the exponent. There are 128 index registers, or B-lines, each 24 bits long; in the instruction code, which is of the one address type, each instruction refers to two B-lines which may modify the address and the average instruction time is between one and two microseconds. There are three control registers, referred to as "main control", "extracode control", and "interrupt control", which are also B-lines 127, 126 and 125. Main control is used by object programs. When main control is active, access to the subsidiary store and V-store is prevented by hardware, and this makes it possible to ensure that object programs cannot interfere with the supervisor. The fixed store contains about 250 subroutines which can be called in from an object program by single instructions called extracodes. When these routines are being obeyed, extracode control is used: extracode control is also used by the supervisor, which requires access to the "private" stores. Interrupt control is used in short routines within the supervisor which deal with peripheral equipment. These routines are entered at times dictated by the peripheral equipments; the program using main or extracode control is interrupted, and continues when the peripheral equipment routine is completed.

The first Atlas installation at Manchester University will include:

16,384 words of core store
8,192 words of fixed store
1,024 words of subsidiary store
98,304 words on drums
8 magnetic tape mechanisms
4 paper tape readers
4 paper tape punches
2 teleprinters
1 line printer
1 card reader
1 card punch

Other Atlas installations will include different amounts of store and peripheral equipments, but the supervisor program herein described is of sufficient generality to handle any configuration through the minor adjustment of parameters.

## 2. THE CO-ORDINATION OF ROUTINES

### The Structure of the Supervisor

The supervisor program controls all those functions of the system that are not obtained merely by allowing the central computer to proceed with obeying an object program, or by allowing peripheral equipments to carry out their built-in operations. The supervisor therefore becomes active on frequent occasions and for a variety of reasons--in fact, whenever any part of the system requires attention from it. It becomes activated in several different ways. Firstly, it can be entered as a direct result of obeying an object program. Thus, a problem being executed calls for the supervisor whenever it requests an action that is subject to control by the supervisor, such as a request for transfer to or from peripheral equipments or the initiation of transfers between core store and magnetic drums; the supervisor is also activated when an object program requires monitoring for any reason such as exponent or division overflow, or exceeding store or time allocation. Secondly, the supervisor may be activated by various items of hardware which have completed their assigned tasks and require further attention. Thus, for example, drums and magnetic tapes call the supervisor into action whenever the transfer of a 512 word block to or from core store is completed; other peripheral equipments require attention whenever the one character or row buffer has been filled or emptied by the equipment. Lastly, certain failures of the central computer store, and peripheral equipments call the supervisor into action.

The central computer thus shares its time between these supervisor activities and the execution of object programs, and the design of Atlas and of the supervisor programs is such that there is mutual protection between object programs and all parts of the supervisor. The supervisor program consists of many branches which are normally dormant but which can be activated whenever required. The sequence in which the branches are activated is essentially random, being dictated by the course of an object program and the functioning of the peripheral equipments.

### Interrupt Routines

The most frequent and rapidly activated parts of the supervisor are the interrupt routines. When a peripheral equipment requires attention, for example, an interrupt flip-flop is set which is available to the central computer as a digit in the V-store; a separate interrupt flip-flop is provided for each reason for interruption. If an interrupt flip-flop is set and interruptions are not inhibited, then before the next instruction is started, the address 2048 of the fixed store is written to the interrupt control register, B125, and control is switched to interrupt control. Further interruptions are inhibited until control reverts to main or extracode control. Under interrupt control, the fixed store program which is held at address 2048 onwards detects which interrupt flip-flop has been set and enters an appropriate interrupt routine in the fixed store. If more than one flip-flop is set, that of highest priority is dealt with first, the priority being built-in corresponding to the urgency of action required. By the use of special hardware attached to one of the B register, B123, the source of any interruption may be determined as a result of obeying between two and six instructions.

The interrupt routines so entered deal with the immediate cause of the particular interrupt. For example, when the one-character buffer associated with a paper tape reader has been filled, the appropriate interrupt flip-flop is set and the "Paper tape reader interrupt routine" is entered. This transfers the character to the required location in store after checking parity where appropriate. The paper tape reader meanwhile proceeds to read the next character to the buffer. Separate interrupt routines in the fixed store control

each type of peripheral equipment, magnetic tapes and drums. The interrupt technique is also employed to deal with certain exceptional situations which occur when the central computer cannot itself deal adequately with a problem under execution, for example, when there is an overflow or when a required block is not currently available in the core store. There are therefore interrupt flip-flops and interrupt routines to deal with such cases. Further routines are provided to deal with interruptions due to detected computer faults.

During the course of an interrupt routine further interruptions are inhibited, and the interrupt flip-flops remain set in the V-store. On resumption of main or extracode control, interruptions are again permitted. If one or more interrupt flip-flops have been set in the meantime, the relevant interrupt routines are obeyed in the sequence determined by their relative priority. In order to avoid interference with object programs or supervisory programs, interrupt routines use only restricted parts of the central computer, namely, the interrupt control register, B-lines 123 and 111 to 118 inclusive, private registers in subsidiary store and the V-store and locked out pages in core store (see Section 3). With the exception of the B-lines, no object program is permitted to use these registers. No lock out is imposed on the B-lines, but interrupt routines make no assumptions concerning the original contents of the B-lines and hence, at worst, erroneous use of interrupt B-lines by an object program can only result in erroneous functioning of that particular program. Switching of control to and from an interrupt routine is rapid, since no preservation of resetting of working registers is required.

The interrupt routines are designed to handle calls for action with the minimum delay and in the shortest time; the character-by-character transfers to and from peripheral equipments, for example, occur at high frequency and it is essential that the transfers be carried out with the minimum possible use of the central computer and within the time limit allowed by the peripheral equipment for filling or emptying the buffer. Since several interrupt flip-flops can become set simultaneously, but cannot be acted upon while another interrupt routine is still in progress, it is essential that a short time limit be observed by each interrupt routine. The majority of calls for interrupt

routines involve only a few instructions, such as the transfer of a character, stepping of counts, etc., and on conclusion the interrupt routine returns to the former control, either main or extracode. On some occasions, however, longer sequences are required; for example, on completion of the input of a paper tape or deck of cards, routines must be entered to deal with the characters collected in the store, writing them to magnetic tape where appropriate, decoding and listing titles and so on. In such cases, the interrupt routine initiates a routine to be obeyed under extracode control, known as a supervisor extracode routine.

### Supervisor Extracode Routines

Supervisor extracode routines (S.E.R.'s) form the principal "branches" of the supervisor program. They are activated either by interrupt routines or by extracode instructions occurring in an object program. They are protected from interference by object programs by using subsidiary store as working space, together with areas of core and drum store which are locked out in the usual way whilst an object program is being executed (see Section 3). They operate under extracode control, the extracode control register of any current object program being preserved and subsequently restored. Like the interrupt routines, they use private B-lines, in this case B-lines 100 to 110 inclusive; if any other working registers are required, the supervisory routines themselves preserve and subsequently restore the contents of such registers. The S.E.R.'s thus apply mutual protection between themselves and an object program.

These branches of the supervisor program may be activated at random intervals. They can moreover be interrupted by interrupt routines, which may in turn initiate other S.E.R.'s. It is thus possible for several S.E.R.'s to be activated at the same time, in the same way as it is possible for several interrupt flip-flops to be set at the same time. Although several S.E.R.'s may be activated, obviously not more than one can be obeyed at any one moment; the rest are either halted or held awaiting execution. This matter is organized by a part of the supervisor called the "co-ordinator routine" which is held in fixed store. Activation of an S.E.R. always occurs via the co-ordinator routine, which arranges

that any S.E.R. in progress is not interrupted by other S.E.R.'s. As these are activated, they are recorded in subsidiary store in lists and an entry is extracted from one of these lists whenever an S.E.R. ends or halts itself. Once started, an S.E.R. is always allowed to continue if it can; a high priority S.E.R. does not "interrupt" a low priority S.E.R. but is entered only on conclusion or halting of the current S.E.R. The co-ordinator has the role of the program equivalent of the "inhibit interrupt flip-flop", the lists of activated S.E.R.'s being the equivalent of the setting of several interrupt flip-flops. The two major differences are that no time limit is placed on an S.E.R., and that an S.E.R. may halt itself for various reasons; this is in contrast to interrupt routines, which observe a time limit and are never halted.

In order that the activity of each branch of the computing system be maintained at the highest possible level, the S.E.R.'s awaiting execution are recorded in four distinct lists. Within each list, the routines are obeyed in the order in which they were activated, but the lists are assigned priorities, so that the top priority list is emptied before entries are extracted from the next list. The top priority list holds routines initiated by completion of drum transfers, and also routines entered as a result of computer failures such as core store parity. The second list holds routines arising from magnetic tape interruptions and the third holds routines arising from peripheral interruptions. The lowest priority list contains one entry for each object program currently under execution, and entry to an S.E.R. through an extracode instruction in an object program is recorded in this list. On completion of an S.E.R., the co-ordinator routine selects for execution the first activated S.E.R. in the highest priority list.

The central computer is not necessarily fully occupied during the course of an S.E.R. The routine may, for example, require the transfer of a block of information from the drum to the core store, in which case it is halted until the drum transfer is completed. Furthermore, the queue of requests for drum transfers (see section 3), is maintained in the subsidiary store, may be full, in which case the S.E.R. making the request must be halted. When an S.E.R. is halted for this or similar reasons, it is returned to the relevant list as halted, and the next activated S.E.R. is entered by the co-ordinator routine. Before an

S.E.R. is halted, a restart point is specified. A halted routine is made free to proceed when the cause of the halt has been removed—for example, by the S.E.R. which controls drum transfers and the extraction of entries from the drum queue. The S.E.R. lists can therefore hold at any one time routines awaiting execution and halted routines; interrupt routines are written in such a way that the number of such S.E.R.'s activated at any one time is limited to one per object program, and one or two per interrupt flip-flop, depending upon the particular features of each interrupt routine. When an S.E.R. is finally concluded, as distinct from halted, it is removed from the S.E.R. lists and becomes dormant again.

Although S.E.R.'s originate in many cases as routines to control peripheral equipment, magnetic tapes and drums, it should not be supposed that this is the sole function of these routines. Entrances to S.E.R.'s from interrupt routines or from extracode instructions in an object program initiate routines which control the entire operation of the computing system, including the transfer of information between store and peripherals, communication with the operators and engineers, the initiation, termination and, where necessary, monitoring of object programs, the monitoring of central computer and peripheral failures, the execution of test programs and the accumulation of logging information. Each branch of supervisory activity is composed of a series of S.E.R.'s, each one activated by an object program or an interrupt routine and terminated usually by initiating a peripheral or magnetic tape transfer or by changing the status of an S.E.R. list or object program list. The most frequently used routines are held in the fixed store; routines required less frequently are held on the magnetic drum and are transferred to core store when required. Supervisor routines in core and drum store are protected from interference by object programs by use of hardware lock-out and the basic store organization routines in the fixed store.

### Object Programs

The function of all supervisor activity is, of course, to organize the progress of problems through the computer with the minimum possible delay. Object programs are initiated by S.E.R.'s, which insert them into the object program list; they are subsequently entered

by the co-ordinator routine effectively as branches of lower priority than any S.E.R. Although object programs are logically sub-programs of the supervisor, they may function for long periods using the computer facilities to the full without reference to the supervisor. For this reason, the supervisor program may be regarded as normally dormant, activated and using the central computer for only a small proportion of the available time.

In order to allow object programs to function with the minimum of program supervision, they are not permitted to use extracode control or interrupt control directly, enabling protection of main programs and supervisor programs to be enforced by hardware. Object programs use the main control register, B127, and are therefore forbidden access to the V-store and subsidiary store. Reference to either of these stores causes the setting of an interrupt flip-flop and hence entrance to the supervisor program.

Access to private stores is only obtained indirectly by use of extracode functions, which switch the program to extracode control and enter one of a possible maximum of 512 routines in the fixed store. These extracode routines form simple extensions of the basic order code, and also provide specific entry to supervisor routines to control the transfer of information to and from the core store and to carry out necessary organization. Such specific entrances to the supervisor program maintain complete protection of the object programs. Protection of magnetic tapes and peripheral input and output data is obtained by the use, in extracode functions, of logical tape and data numbers which the supervisor identifies within each program with the titles of the tapes or information. Blocks of core and drum store are protected by hardware and by the supervisor routines in fixed store as described in Section 3.

An object program is halted (by S.E.R.'s) whenever access is required to a block of information not immediately available in the core store. The block may be on the drums, in which case a drum transfer routine is entered, or it may be involved in a magnetic tape transfer. In both cases the program is halted until the block becomes available in core store. In the case of information involved in peripheral transfers, such as input data or output results, the supervisor buffers the information in core and drum store, and

"direct" control of a peripheral equipment by an object program is not allowed. In this way, immobilization of large sections of store whilst a program awaits a peripheral transfer can be avoided. A program may however call directly for transfers involving drums or magnetic tapes by use of extracode functions, which cause entrance to the relevant supervisor routines. Queues of instructions are held in subsidiary store by these routines, in order to allow the object program to continue and to achieve the fullest possible overlap between tape and drum transfers and the execution of an object program.

While one program is halted, awaiting completion of a magnetic tape transfer for instance, the co-ordinator routine switches control to the next program in the object program list which is free to proceed. In order to maintain full protection, it is necessary to preserve and recover the contents of working registers common to all programs such as the B-lines, accumulator, and control registers, and to protect blocks in use in core store. The S.E.R. to perform this switching from one object program to another occupies the central computer for around  $750 + 12p \mu\text{secs}$ , where  $p$  is the number of pages, or 512 word blocks in core store. On the Manchester University Atlas, which has 32 pages of core store, the computing time for the round trip to switch from one program to another and to return subsequently is around 2.5 m.secs. This is in contrast to the time of around  $60 \mu\text{secs}$ . to enter and return from an S.E.R. and even less to switch to and from an interrupt routine. It is therefore obvious that the most efficient method of obtaining the maximum overlap between input and output, magnetic tape transfers, and computing is to reduce to a minimum the number of changes between object programs and to utilize to the full the rapid switching to and from interrupt and supervisor routines. The method of achieving this in practice is described in Section 6.

Compilation of programs is treated by the supervisor as a special case of the execution of an object program, the compiler comprising an object program which treats the source language program as input data. Special facilities are allowed to compilers in order that their allocation of storage space may be increased as need arises, and to allow exit to the supervisor before the execution of

a problem or the recording of a compiled object program.

### Error Conditions

In addition to programmed entrances to the supervisor, entrance may also be made in the event of certain detectable errors arising during the course of execution of a problem. A variety of program faults may occur and be detected by hardware, by programmed checks in extracodes, and in the supervisor. Hardware causes entry to the supervisor by the setting of interrupt flip-flops in the event of overflow of the accumulator, use of an unassigned instruction, and reference to the subsidiary store or V-store. Extracode routines detect errors in the range of the argument in square root, logarithm, and arcsin instructions. In the extracodes referring to peripheral equipment or magnetic tapes, a check is included that the logical number of the equipment has been previously defined. In extracodes for data translation, errors in the data may be detected. The supervisor detects errors in connection with the use of the store. All problems must supply information to the supervisor on the amount of store required, the amount of output, and the expected duration of execution. This information is supplied before the program is compiled, or may be deduced after compilation. The supervisor maintains a record of store blocks used, and can prevent the program exceeding the preset limit. In addition, an interrupt flip-flop is set by a clock at intervals of 0.1 secs, and another flip-flop is set whenever 1024 instructions have been obeyed using main or extracode control. These cause entrances to the supervisor which enable a program to be "monitored" to ensure that the preset time limit has not expired, and which are also instrumental in initiating routines to carry out regular timed operations such as logging of computer performance and initiation of routine test programs.

The action taken by the supervisor when a program "error" is detected depends upon the conditions previously set up by the program. Certain errors may be individually trapped, causing return of control to a preset address; a private monitor sequence may be entered if required enabling a program or a compiler to obtain diagnostic printing; failing specification of these actions, some information is printed by the supervisor and the

program is suspended, and usually dumped to magnetic tape to allow storage space for another program.

The following sections describe in detail the action of certain supervisor routines, namely, those controlling drums, magnetic tapes, and peripheral equipment and those controlling the flow of information in the computer.

### 3. STORE ORGANIZATION

#### Indirect addressing and the One-Level Store

The core store of Atlas is provided with a form of indirect addressing which enables the supervisor to re-allocate areas of store and to alter their physical addresses, and which is also used to implement automatic drum transfers. With each page, or 512 word block, of core store there is associated a "page address register" which contains the most significant address bits of the block of information contained in the page. Every time access is required to a word of information in the core store, the page containing the word is located by hardware. This tests for equivalence between the requested "block address", or most significant address bits, and the contents of each of the page address registers in parallel. Failure to find equivalence results in a "non-equivalence" interruption. The page address registers are themselves addressable in the V-store and can thus be set appropriately by the supervisor whenever information is transferred to or from core store.

One of the most important consequences of this arrangement is that it enables the supervisor to implement automatic drum transfers. The address in an instruction refers to the combined core and drum store of the computer, and the supervisor records in subsidiary store the location of each block of information; only one copy of each block is kept, and the location is either a page of core store or a sector of the drum store. At any moment, only some of the blocks comprising a particular program may be in the core store and if only these blocks are required, the program can run at full speed. When a block is called for which is not in the core store, a non-equivalence interruption occurs, which enters the supervisor to transfer the new block from a sector of the drum to a

page of the core store. During this operation the program that was interrupted is halted by the supervisor.

The block directory in subsidiary store contains one entry for each block in the combined core and drum store. It is divided into areas for each object program which is in the store; a separate program directory defines the area of the block directory occupied by each program. The size of this area, or the number of blocks used by a program, is specified before the program is obeyed in the job description (see Section 6). The entry for block  $n$  contains the block number  $n$  together with the number of the page or sector occupied by the block, and, if possible, is made in the  $n^{\text{th}}$  position in the area; otherwise the area is filled working backwards from the end. In this way, blocks used by different object programs are always kept distinct, regardless of the addresses that are used in each program. A program addresses the combined "one-level store\*" and the supervisor transfers blocks of information between the core and drum store as required; the physical location of each block of information is not specified by the program, but is controlled by the supervisor.

There are occasions when an object program must be prevented from obtaining access to a page of the core store, such as one involved in a drum or tape transfer. To ensure complete protection of such pages, an additional bit, known as a lock out bit, is provided with each page address register. This prevents access to that page by the central computer, except when on interrupt control, and any reference to the page causes a non-equivalence interruption. By setting and re-setting the lock out bits, the supervisor has complete control over the use of core store; it can allow independent object programs to share the core store, it can reserve pages for peripheral transfers and can itself use parts of the core store occasionally for routines or working space, without any risk of interference. This is done by arranging that, whenever control is returned to an object program, pages that are not available to it are locked out.

\*A paper on the one-level store has been written, and will, it is hoped, be published by the Institute of Radio Engineers.

A block of information forming part of an object program may also be locked out from use by that program because an operation on that information, controlled by the supervisor, is not complete. A drum, magnetic tape, or peripheral equipment transfer involving this block may have been requested. The reason for the lock out of such a block is recorded in the block directory, and if the block is in the core store, the lock out digit is also set. If reference is made to such a block by the object program, a non-equivalence interruption occurs and a supervisor extracode routine halts the program. This S.E.R. is restarted by the co-ordinator routine when the block becomes "unlocked", and the object program is re-entered when the block is available in core store.

#### The Drum Transfer Routine

The drum transfer routine is a group of S.E.R.s which are concerned with organizing drum transfers, and updating page address registers and the block directory. Once initiated, the transfer of a complete block to or from the drum proceeds under hardware control; the drum transfer routine initiates the transfer and identifies the required drum sector by setting appropriate bits in the V-store. It also identifies the core store page involved by setting a particular "dummy" block address, recognized by the drum control hardware, in the page address register; at the same time, this page is locked out to prevent interference from object programs while the transfer is in progress.

On completion of a transfer, an interruption occurs which enters the drum transfer routine. The routine can also be entered from the non-equivalence interrupt routine, which detects the number of the block requested but not found in the page address registers. Finally, the drum transfer routine can be activated by other parts of the supervisor which require drum transfers, and by extracode instructions which provide a means whereby object programs can if they wish exert some control over the movement of blocks to and from the drum store. A queue of requests for drum transfers, which can hold up to 64 requests, is stored in the subsidiary store; when the drum transfer routine is entered on completion of a transfer, the next transfer in the queue is initiated.

Whenever the supervisor wishes to enter another request for a drum transfer, three possible situations arise. Firstly, the queue is empty and the drum transfer can be started immediately. Secondly, the queue is already partly filled and the request is entered in the next position in the queue. Thirdly, the queue is full. In this case the routine making the request is halted by the co-ordinator routine, and is resumed when the queue can receive another entry. In the first two cases the supervisor routine is concluded when the request reaches the queue.

A non-equivalence interruption, which implies a drum transfer is required, is dealt with as follows. The core store is arranged to always hold an empty page with no useful information in it, and when required, a transfer of a block of information from the drum to this empty page is initiated. While this drum transfer is proceeding, preparation is made to write up the contents of another page of core store to the drum to maintain an empty page. The choice of this page is the task of the "learning program" which keeps details of the use made of blocks of information. This learning program will be described in detail elsewhere; it predicts the page which will not be required for the largest time, and is arranged with a feed-back so that if it writes up a block which is almost immediately required again, it only does this once. The number of the chosen page

the drum queue entry is converted to a request to write this page to the drum. This supervisor routine is now concluded and returns control to the co-ordinator routine.

When the drum transfer is completed, the drum transfer routine is again entered. This updates the block directory and page address register, makes the object program free to proceed and initiates the next drum request, which is to write the chosen page to the drum. This routine is now concluded and the co-ordinator is re-entered. The supervisor is finally entered when the write to drum transfer is complete. The block directory is updated, a note is made of the empty page, and the next drum request is initiated.

#### The Use of Main Store by the Supervisor

Some routines of the supervisor are obeyed in the main store, and these and others use working space in the main store. Since the supervisor is entered without a complete

program change, special care must be taken to keep these blocks of store distinct and protected from interference. The active supervisor blocks of main store are recorded in the area for program 0 in the block directory. There are also some blocks of the supervisor program which are stored permanently on the drum; when one of these permanent blocks is required, it is duplicated to form an active block of the supervisor or, as in the case of a compiler, to become part of an object program.

Of the possible 2048 block numbers, 256 are "reserved" block numbers which are used exclusively by the supervisor and are not available to object program; object program are restricted to using the remaining "non-reserved" block numbers. Blocks with reserved block numbers may be used in the core store at any time by the supervisor, and the co-ordinator routine locks out these pages of core store before returning control to an object program. The supervisor also uses some blocks having non-reserved block numbers to keep a record of sequence of blocks of information such as input and output streams. When a non-reserved supervisor block is called to the core store, the page address register is not set, since there may be a block of an object program which has the same block number already in the core store. Instead, the page address register is set to a fixed reserved block number while it is in use, and is cleared and locked out before control passes to another routine.

Not all the reserved block numbers are available to the supervisor for general use, since certain block numbers are temporarily used when drum, tape, and peripheral transfers are proceeding. These block numbers do not appear in the block directory. For example, when a magnetic tape transfer is taking place, the page of core store is temporarily given a block number which is recognized by the hardware associated with that tape channel. When the transfer is complete, the appropriate block number is restored. During a peripheral transfer, and also on other occasions, it is necessary that a block should be retained in the core store and should not be transferred to the drum. The relevant page of core store is "locked down" by setting a digit in the subsidiary store; the learning program never selects for transfer to the drum a page for which this lock-down digit is set.

#### 4. MAGNETIC TAPE SUPERVISOR ROUTINES

##### The Magnetic Tape Facilities

The tape mechanism used on Atlas is the Ampex TM2 (improved FR 300) using one inch wide magnetic tape. There are sixteen tracks across the tape - twelve information tracks, two clock tracks, and two tracks used for reference purposes. The tapes are used in a fixed-block, pre-addressed mode. Information is stored on tape in blocks of 512 forty-eight bit words, together with a twenty-four bit checksum with end around carry. Each block is preceded by a block address and block marker and terminated by a block marker; the leading block address is sequential along the tape, and what is effectively the trailing block address is always zero. Tapes are tested and pre-addressed by special routines before being put into use, and the fixed position of the addresses permits selective overwriting and simple omission of faulty patches on the tape. Blocks can be read when the tape is moving either in the forward or reverse direction, but writing is only possible when the tape is moving forward. The double read and write head is used to check read when writing on the tape. When not operating the tape stops with the read head midway between blocks.

Atlas may control a maximum of 32 magnetic tape mechanisms. Each mechanism is connected to the central computer via one of eight channels, all of which can operate simultaneously, each controlling one read, write or positioning operation. It is possible for each tape mechanism to be attached to either one of a pair of channels, the switching being under the control of supervisory program through digits in the V-store. Fast wind and rewind operations are autonomous and only need the channel to initiate and, if required, terminate them. Transfer of a 512-word block of information between core store and tape is effected via a one-word buffer, the central computer hesitating for about  $1/2 \mu\text{sec}$ , on average, each time a word is transferred to or from the core store. During a transfer the page of core store is given a particular reserved block number and the contents of the page address register are restored at the end of the transfer.

Supervisory programs are only entered when the block addresses are read before and after each block, and when the tape stops. As each block address is read, it is recorded in the V-store and an interrupt flip-flop is set, causing entrance to the block address interrupt routine.

##### The Block Address Interrupt Routine

This routine is responsible for initiating and checking the transfer of a single block between tape and core store, and searching along the tape for a specified block address. Digits are available in the V-store to control the speed and direction of motion of the tape and the starting and termination of read or write transfers. The block addresses are checked throughout and, in particular, a write transfer is not started until the leading block address of the tape block involved has been read and checked. Hardware checking is provided on all transfers, and is acted upon by supervisor routines. A 24-bit check sum is formed and checked as each block is transferred to or from a tape, and a digit is set in the V-store if any failure is detected. Similarly a digit is set in the event of failure to transfer a full block of 512 words. These digits are tested by the block address interrupt routine on the conclusion of each transfer. Parity failure either on reading from core store or on formation of the parity during a transfer to core store causes the setting of interrupt flip-flops. If a tape fails to stop, this is detected by the block address interrupt routine as a particular case of block address failure. Failure to enter the block address routine (for example, through failure to read block markers) is detected by the timed interrupt routine at intervals of 100 milliseconds. Finally, failures of the tape mechanism, such as vacuum failure, set a separate interrupt flip-flop. The detection of any of these errors causes entry to tape monitor routines, whose action will be described later.

##### Organization of Tape Operations

Magnetic tape operations are initiated by entrance to the tape supervisor routines in the fixed store from extracode instructions in an object program or, if the supervisor requires the tape operation for its own purposes, from supervisor extracode routines. From a table in subsidiary store, the logical tape number

used in a program is converted to the actual mechanism number, and the tape "order" is entered to a queue of such orders, in subsidiary store, awaiting execution. A tape order may consist of the transfer of several blocks and any store blocks involved are "locked out" to prevent subsequent use before completion of the transfers; if any block is already involved in a transfer, the program initiating the request is halted. Similarly, the program is halted if the queue of tape instructions is already full. If the channels to which the deck can be connected are already occupied in a transfer or positioning, the tape supervisor returns control to the object program, which is then free to proceed. A program may thus request a number of tape transfers without being halted, allowing virtually the maximum possible overlap between the central computer and the tape mechanisms during execution of a program. Should a channel be available at the time a tape order is entered to the queue, the order is initiated at once by writing appropriate digits to the V-store, and by writing reserved tape transfer block numbers to the appropriate page address registers if the order involves a read or write transfer. The tape supervisor then returns control to the object program or supervisor routine.

One composite queue of tape orders is used for orders relating to all tape mechanisms and orders are extracted from the queue by S.E.R.'s entered from the block address interrupt routine. On reading the penultimate block address involved in an operation (for example, the last leading block address in a forward transfer) the next operation for the channel is located, and if it involves the same mechanism as the current order, and tape motion in the same direction, the operation is "prepared" by calling any store block involved to core store. On reading the final block address and successfully concluding checks, the block address interrupt routine initiates the next operation immediately if one has been prepared, thus avoiding stopping the tape if possible. If no operation has been prepared, the interrupt routine stops the tape by setting a digit in the V-store, and a further "block address interruption" occurs when the tape is stopped and the channel can accept further orders. This interruption enters an S.E.R. which extracts the next order for the channel from the tape queue, and the cycle of events is repeated until no further order for

this channel remains. As each transfer is concluded, any object program halted through reference to the store block is made free to proceed.

An exception to the above process is when a long movement (over 200 blocks) or a rewind is required. In this case, the movement is carried out at fast speed, with block address interruptions inhibited, and the channel may meanwhile be used to control another tape mechanism. The long movement is terminated by checking the elapsed time and at the appropriate moment, entering the tape supervisor from the timed interrupt routine. The mechanism is then brought back "on channel" and the speed is returned to normal. When reading of block addresses is correctly resumed, the search is continued in the normal manner.

#### The Title Block

The first block on each magnetic tape is reserved for use by the supervisor, and access to information in this block by an object program is through special instructions only. This block contains the title of the tape, or an indication that the tape is free. When magnetic tapes are required by the supervisor or by an object program, the supervisor prints instructions to the operator to load the named tape and to engage the mechanism on which it is loaded. The engage button of each mechanism (see section 5) is attached to a digit in the V-store, and these digits are scanned by the supervisor every one second. When a change to "engaged" status has been detected, the tape supervisor is entered to read the first block from the tape. The title is then checked against the expected title. In this way, the presence of the correct tape is verified, and furthermore the tape bearing the title becomes associated with a particular mechanism. Since the programmer assigns a logical tape number to the tape bearing a given title, this logical tape number used in extracode instructions can be converted by the supervisor to the actual mechanism number. Other supervisory information is included in the first block on each tape, including a system tape number and the number of blocks on the tape. Special supervisory routines allow Atlas to read tapes produced on the Ferranti Orion computer, which used the same tape mechanisms but can write blocks of varying lengths on the tape. These tapes are distinguished on Atlas by a marker written in the title block.

## Magnetic Tape Failures

All failures detected by the interrupt routines cause the block address interrupt routine to stop the tape at the end of the current block when possible, and then to enter tape monitor supervisory routines; if the tape cannot be stopped, it is disengaged and the tape monitor routines entered. These routines are S.E.R.'s designed to minimize the immediate effect on the central computer of isolated errors in the tape system, to inform maintenance engineers of any faults, and to diagnose as far as possible the source of a failure. As an example of the actions taken by monitor routines, suppose a check sum failure has been detected while reading a block from tape to core store. The tape monitor routines make up to two further attempts to read the block; if either succeeds, the normal tape supervisor is re-entered after informing the engineers. Repeated failure may be caused by the tape or the tape mechanism; to distinguish these, the tape is rewound and an attempt is made to read the first block. If this is successful, a tape error is indicated, and an attempt is made to read the suspect block with reduced bias level. Failure causes the mechanism to be disengaged and the program using the tape to be suspended. If the "recover read" is successful, the tape is copied to a free tape and the operators instructed to re-address the faulty tape, omitting the particular block which failed. If on rewinding the tape, the first block cannot be read successfully, failure in the tape mechanism is suspected and the operator is instructed to remount the tape on another mechanism. Other faults are monitored in a similar manner, and throughout, the operator and engineers are informed of any detected faults. Provision is made for the program using the tape to "trap" persistent tape errors and thereby to take action suitable to the particular problem, which may be more straight-forward and efficient than the standard supervisory action.

Addressing of new tapes and re-addressing of faulty tapes are carried out on the computer by supervisory routines called in by the operator. A tape mechanism is switched to "addressing mode", which prohibits transfers to and from the core store, permits writing from the computer to the reference tracks and to the block addresses on tape,

and activates a timing mechanism to space the block addresses. When a new tape is addressed, addresses are written sequentially along the tape and the area between leading and trailing block addresses is checked by writing ones to all digit positions and detecting failures on reading back. Any block causing failure is erased and the tape spaced suitably. On completion, a special block address is written to indicate "end of tape" and the entire tape is then checked by reading backwards. Any failures cause entry to the re-addressing routine. Finally, the tape mechanism is returned to "normal" mode, a little block is written containing the number of blocks on tape, a tape number, and the title "Free", and the tape is made available for use. A tape containing faulty blocks is re-addressed, omitting such blocks, by entry to the re-addressing routine with a list of faulty blocks; the faulty blocks are erased and the remaining blocks are re-labelled sequentially, the tape being checked as when addressing a new tape.

## 5. PERIPHERAL EQUIPMENT

### Peripheral Interruptions

As mentioned in the Introduction, a large number and variety of peripheral equipments may be attached to Atlas. However, the amount of electronics associated with each equipment is kept to a minimum, and use is made of the high computing speed and interruption facilities of Atlas to provide control of these equipments and large scale buffering.

Thus the paper tape readers, which operate at 300 characters per second, set an interrupt flip-flop whenever a new character appears. (Characters may be either 5 or 7 bits depending on which of two alternative widths of tape is being read.) Similarly the paper tape punches, and the teleprinters which print information for the computer operators, cause an interruption whenever they are ready to receive a new character; these equipments operate at 110 and 10 characters per second respectively.

The card readers read 600 cards per minute, column by column, and interrupt the computer for every column. The card punches at 100 cards per minute, punch by rows and interrupt for each row.

The printers, which have 120 print wheels bearing 50 different characters, cause an

interruption as each character approaches the printing position so that the computer may prime the hammers for those wheels where this character is to be printed. There are therefore 50 interruptions per revolution, or one every 1-1/2 milliseconds.

All the information received from, or sent to, these peripheral equipments does so via particular digit positions in the V-store. For example, there are 7 such bits for each tape reader, and 120 for each printer, together with a few more bits for control signals.

The majority of interruptions can be dealt with simply by the interrupt routine for the particular type of equipment. Thus the paper tape reader interrupt routine normally has merely to refer to a table of characters to apply code conversion and parity checks, and to detect terminating characters and if all is well to store the new character in the next position in the store.

The interrupt routines for the printers and card punches are not expected to do the conversion from character coding to row binary; this is done by an S.E.R. before the card or line of print is commenced. The card routines are however complicated by the check reading stations; punching is checked one card cycle afterwards, and reading is checked 3 columns later. The interrupt routines apply these checks, and in the event of failure a monitor S.E.R. is entered.

The printer interrupt routine counts its interruptions to identify the character currently being printed; a check is provided once each revolution when a datum mark on the printer shaft causes a signal bit to appear in the V-store. This counting is maintained during the paper feed between lines, which occupies several milliseconds.

#### Attention by Operators

Whenever equipment needs attention it is "disengaged" from the computer. In this state, which is indicated by a light on the equipment and a corresponding bit in the V-store, it automatically stops and cannot be started by the computer.

The operator may engage or disengage an equipment by means of two buttons so labelled. The equipment may also be disengaged by the computer by writing to the appropriate V-store bit, but the computer cannot engage it.

The "engage" and "disengage" buttons do not themselves cause interruptions of the central computer. Instead, the "engaged" bits in the V-store are examined every second (this routine is activated by the clock interruption) and any change activates the appropriate S.E.R. Disengaging a device does not inhibit its interruptions, so that if the operator disengages a card machine in mid-cycle to replenish the magazine or to empty the stacker, the cycle is completed correctly.

There are also other special controls for particular equipments, e.g., a run-out key on card machines, and a 5/7-hole tape width selector switch on punched tape readers.

Most devices have detectors that indicate when cards or paper are exhausted or running low. These correspond to bits in the V-store that are read by the appropriate S.E.R. The paper tape readers however have no such detector, and the unlikely event of a punched tape passing completely through a reader (due to the absence of terminating characters) appears to the computer merely as a failure to encounter a further character within the normal time interval. This condition is detected by the one-second routine.

#### Store Organization of Input and Output Information

In general, input information is converted to a standard 6-bit internal character code by the interrupt routine concerned, and placed in the store 8 characters to a word. (An exception to this occurs in the case of card readers when they are reading cards not punched in a standard code, in which case the 12 bits from one column are simply copied into the store and occupy two character positions. A similar case is 7-hole punched tape, when this is used to convey 7 information bits without a parity check. Such information is distinguished by warning characters, both on the input medium and in the store.)

A certain amount of supervisor working space in the core store is set aside to receive this information from the interrupt routines, and is subdivided between the various input peripherals. The amount of this space depends on the number and type of peripherals attached; the first two Atlases will normally use one block (512 words). This block will be locked down in a page of the core store whenever any input peripheral is operating (i.e., most of the time).

As each input equipment fills its share of this block, the information is copied by an S.E.R. into another block devoted exclusively to that equipment. These copying operations are sufficiently rare that the latter block need not remain in core store in the meantime; in fact it is subject to the same treatment as object programs by the drum transfer routine, and may well be put onto a drum and brought back again for the next copying operation. Thus only one page of core store is used full time during input operations, but nevertheless each input stream finds its way into a separate set of blocks in the store.

The page that is shared between input peripherals is subdivided in such a way as to minimize the number of occasions on which information must be copied to other blocks; it turns out that the space for each equipment needs to be roughly proportional to the square root of its information rate.

Similarly, information intended for output is placed in a common output page, subdivided for the various output devices, and is taken from there by the interrupt routines as required. The interrupt routines for teleprinters and tape punches do the necessary conversion from the internal character code used by the device. As soon as the information for a particular device is exhausted, an S.E.R. is activated to copy fresh information into the common output page. Again, the page is subdivided roughly in proportion to the square roots of the information rates.

For card punches and printers, whose interrupt routines require their information arranged in rows of bits, a further stage of translation is necessary. In these cases, on completing a card or line, internal 6-bit characters are converted by an S.E.R. into a card or line image also in the output block. In fact, the desirable amount of working space for output buffering somewhat exceeds one block, and the spare capacity of the subsidiary store will be utilized to augment it.

## 6. THE OPERATING SYSTEM

The following is a synopsis of work explained in detail in a paper the Computer Journal.

### Input

The fast computing speed of Atlas and the use of multiple input and output peripheral

equipments enable the computer to handle a large quantity and variety of problems. These will range from small jobs for which there is no data outside the program itself, to large jobs requiring several batches of data, possibly arriving on different media. Other input items may consist of amendments to programs, or requests to execute programs already supplied. Several such items may be submitted together on one deck of cards or length of punched tape. All must be properly identified for the computer.

To systematize this identification task, the concept of a "document" has been introduced. A document is a self-contained section of input information, presented to the computer consecutively through one input channel. Each document carries suitable identifying information (see below) and the supervisor keeps in the main store a list of the documents as they are accepted into the store by the input routines, and a list of jobs for which further documents are awaited.

A job may require several documents, and only when all these have been supplied can execution begin. The supervisor therefore checks the appearance of documents for each job; when they are complete the job scheduling routine is notified (see below)

Normally, the main core and drum store of the computer is unlikely to suffice to hold all the documents that are waiting to be used. The blocks of input information are therefore copied, as they are received, onto a magnetic tape belonging to the supervisor, called the "system input tape." Hence, if it becomes necessary for the supervisor to erase them from the main store, they can be recovered from the system input tape when the job is ready for execution.

The system input tape thus acts as a large scale buffer, and indeed it plays a similar part to that of the system input tape in more conventional systems. The differences here are that the tape is prepared by the computer itself instead of by off-line equipment, and that there is no tape-handling of manual supervision required after the input of the original documents - an important point in a system designed to handle many miscellaneous jobs.

This complete bufferage system for input documents is called the "input well." Documents awaiting further documents before they can be used are said to be in "input well A"; complete sets of documents for jobs from "input well B." Usually documents being

accepted into input well B must be read from the system input tape back into the main store so that they are ready for execution; often however they will already be in input well A in the main store, so that only an adjustment of the block directory is required.

One result of this arrangement is that the same tape is being used both to write input blocks, in a consecutive sequence, and to read back previously written blocks to recover particular documents as they are required. The tape will therefore make frequent scans over a few feet of tape, although it will gradually progress forwards. The lengths of these scans are related to the main store space occupied by input well A. For example, so long as the scans do not exceed about 80 feet (130 blocks) the waiting time for writing fresh blocks will remain less than the time for input of three blocks from a card reader, so that comparatively little main store space need be occupied by input well A. To ensure that scans are kept down to a reasonable limit, any documents left on the system input tape for so long that they are approaching the limit of the scannable area are copied to the system dump tape (see below). If the number of these becomes large, the computer operators are warned to reduce the supply of documents through the input peripherals.

### Output

The central computer can produce output at a much greater rate than the peripheral equipments can receive it, and an "output well" is used in a manner analogous to the input well. This well uses a "system output tape" to provide bulk buffering.

Output for all output peripherals is put onto the same tape, arranged in sections that are subdivided so that the contents of a section will occupy all currently operating peripherals for the same length of time. Thus if, for example, a burst of output is generated for a particular peripheral, it is spaced out on the system output tape, leaving spare blocks to be filled in later with output for other peripherals (this is possible because Atlas uses pre-addressed tape). In this way, the recovery of information from the tape into "output well B" as required by the various peripherals merely involves reading complete sections from the tape.

Again, there is a limit to the amount of information that can usefully be buffered on

the output tape, due to the time required to scan back and forth between writing and reading regions, and this limit depends on the space available in the main store for output well B. An S.E.R. keeps a check on the amount of information remaining in output well B for each equipment, and relates this to the present scan distance to decide when to start to move the tape back for the next reading operation. If the amount of output being generated by object programs becomes too great some of it is put instead on the dump tape (see below) or a program is suspended.

### The System Dump Tape

The system input and output tapes operate essentially as extensions of the main store of the computer. Broadly speaking, documents are fed into the computer, programs are executed, and output is produced. The fact that the input and output usually spends some time on magnetic tape is, in a sense, incidental. This input and output buffering is, however, a continuous and specialized requirement, so that a particular way of using these tapes has been developed and special S.E.R.'s have been written to control them.

When demands on storage exceed the capacity of the main store and input and output tapes, a separate magnetic tape, the system dump tape, is used to hold information not required immediately. This tape may be called into use for a variety of reasons. Execution of a problem may be suspended and the problem recorded temporarily on the dump tape if other problems are required to fill the output well, or alternatively if its own output cannot be accommodated in the output well. Also, as already described, the input and output wells can "overflow" to the system dump tape. This tape is not used in a systematic manner, but is used to deal with emergencies. However, the system is such that, if necessary, the system input and output tapes can be dispensed with, thereby reducing the input and output wells and increasing the load on the system dump tape. In an extreme case, the system dump tape itself can be dispensed with, implying a further reduction in the efficiency of the system.

### Headings and Titles

Every input document is preceded by its identifying information, mentioned above.

This consists of two lines of printing, forming the heading and the title respectively.

The heading indicates which type of document follows. The most common headings are

COMPILER followed by the name of a program language, which means that the document is a program in the stated language;

DATA which means that the document is data required by an object program; and

JOB which means that the document is a request for the computer to execute a job, and gives some relevant facts about it.

The last type of document is called a "job description." It gives, for example, a list of all other documents required for the job, a list of output streams produced, any magnetic tapes required and upper limits to the storage space and computing time required. Many of these details are optional; for example if storage space and computing time are not quoted a standard allowance will be made.

For example, if a program operates on two data documents which it refers to as data 1 and data 2, the job description would contain:

#### INPUT

- 1 followed by the title of data 1
- 2 followed by the title of data 2

The program would appear in this list as data 0. Alternatively, a job description may be combined with a program, forming one composite document, and this will usually happen with small jobs.

Each output stream may be assigned to a particular peripheral or type of peripheral, or may be allowed to appear on any output equipment. The amount of output in each stream may also be specified. For example, a job description may include:

#### OUTPUT

- 1 LINE PRINTER 20 BLOCKS
- 2 CARDS
- 3 ANY

Each magnetic tape used by a program is identified by a number within the program, and the job description contains a list of these numbers with the title that appears in block 0 of each tape to identify it; for example:

#### TAPE

- 1 POTENTIAL FIELD CYLIND/204-TPU5.

If a new tape is required, a free tape must be loaded, which the program may then adopt and give a new title. This is indicated thus:

#### TAPE FREE

- 2 MONTE CARLO RESULTS K49-REAC-OR4.

The loading of tapes by operators is requested by the supervisor acting on the information in job descriptions.

Finally, the end of a document is indicated by

\* \* \*

and if this is also the end of the punched tape or deck of cards it is followed by the letter Z. On reading this the computer disengages the equipment.

#### Logging and Charging for Machine Time

As problems are completed, various items of information on the performance of the computing system are accumulated by the supervisor. Items such as the number of program changes and the number of drum transfers are accumulated and also, for each job, the number of instructions obeyed, the time spent on input and output, and the use made of magnetic tapes. These items are printed in batches to provide the operators with a record of computer performance, and they are also needed for assessing machine charges.

The method of calculating charges may well vary between different installations, but one desirable feature of any method is that the charge for running a program should not vary significantly from one run to another. One difficulty is that the number of drum transfers required in a program may vary considerably with the amount of core store which is being used at the same time for magnetic tape and peripheral transfers. One method of calculating the charge so as not to reflect this variation is to make no charge for drum transfers, but to base the charge for computing time on the number of instructions obeyed in a program. This, however, gives no incentive to a programmer to arrange a program so as to reduce its drum transfers, and more elaborate schemes may eventually be devised. The charge for using

peripherals for input and output can be calculated from the amount of input and output. For magnetic tapes, the charge can be based on the length of time for which the tape mechanism is engaged, allowance being made for the time when the program is free to proceed but is held up by a program of higher priority. All this information is made available to the S.E.R. responsible for the costing of jobs.

#### Methods of Using the Operating System

The normal method of operating the computer is for documents to be loaded on any peripheral equipment in any order, although usually related documents will be loaded around the same time. The titles and job descriptions enable the supervisor program to assemble and execute complete programs, and the output is distributed on all the available peripherals. Usually programs are compiled and executed in the same order as the input is completed, but the supervisor may vary this depending on the load on different parts of the system. For example, a problem requiring magnetic tape mechanisms which are already in use may be by-passed in favour of a problem using an idle output peripheral; a problem which computes for a long time may be temporarily suspended in order to increase the load on the output peripherals. By these and similar methods, the S.E.R. responsible for scheduling attempts to maintain the fullest possible activity of the output peripherals, the magnetic tape mechanisms and the central computer.

Documents may also be supplied to the computer from magnetic tapes; these tapes may be either previous system input tapes or library tapes or tapes on which "standard", frequently used, programs are stored. Such documents are regarded as forming part of Input Well B and are read into main store when required. An alternative method of operating may be to use the computer to copy documents to a "private" magnetic tape, rather than to use the system input tape, and at a later time to supply the computer with a succession of jobs from this tape. Similarly, output may be accumulated on a private magnetic tape and later passed through the computer to one or more peripheral equipments. Routines forming part of the supervisor are available to carry out such standard "copying operations.

Provision is also made for the chief operator to modify the system in various ways;

for example, priority may be given to a particular job, or a peripheral equipment may be removed from general use and allocated a particular task. An "isolated" operating station may, for example, be established by reserving a particular output equipment for use by problems loaded on a particular input equipment.

#### 7. Conclusion

The Atlas supervisor program is perhaps the most advanced example so far encountered of a program involving many parallel activities, all closely interconnected. Although a great deal of it has already been coded at the time of writing, there are still a few details to be thrashed out, and no doubt many changes will have to be made to suit conditions existing at various installations. The overall structure of the program is therefore of prime importance: only if this structure is adequate, sound and systematic will it be possible to complete the coding satisfactorily and to make the necessary changes as they arise.

The structure described in this paper, involving interrupt routines and "supervisory extracode routines" controlled by a coordinating routine, has proved eminently satisfactory as a basis for every supervisory task that has so far been envisaged, and it is expected that all variations that might be called for will fit into this structure.

There is no doubt that its success as a supervisory scheme for a very fast computer is due largely to certain features of the Atlas hardware, in particular the provisions for protecting programs from interference, and the page address registers. The way in which the latter permit information to be moved around at run time without the need to recompile programs gives the supervisor an entirely new degree of freedom. The possible uses of this facility have turned out to be so extensive that the task of utilizing such a large computer effectively without it seems by comparison to be almost impossible.

#### Acknowledgments

The work described in this paper forms part of the Atlas project by a joint team of Manchester University and Ferranti Ltd., whose permission to publish is acknowledged. The authors wish to express their appreciation of the assistance given by the other members of the Atlas team.

# *Ferranti Ltd*

## COMPUTER DEPARTMENT

*Enquiries to*

*London Computer Centre, 68 NEWMAN STREET, LONDON, W.1*  
Telephone MUSEum 5040

*and*

21 PORTLAND PLACE, LONDON, W.1  
Telephone LANgham 9211

*Office, Works and Research Laboratories*

WEST GORTON, MANCHESTER, 12. Telephone EAST 1301

*Research Laboratories*

WESTERN ROAD, BRACKNELL, BERKS