c) characters must be six-bit, internal code, and must be packed eight to the word.

Example:

The text stored by L100 in the example at the end of paragraph 8.6.6 could be printed on output 4 by the instructions:-

| | | | | |
|---|---|---|---|---|
| 1060 | 0 | 0 | 4 | Select output 4 |
| 121 | 89 | 0 | 100 | b89' = 100 |
| 1362 | 0 | 0 | A6/L1 | Print text. |

8.10.7 A7/L1

Entry to A7/L1 will cause the output of a text in the same way as entry to A6/L1 except that the carriage control character terminating the (last) record of the text is replaced by a carriage control character specified before entry in B87.

In particular, if further information is to follow on the same line as the last line of the text, entry should be to A7/L1 with B87 zero. The last line of text will not then be terminated.





8.11 Optional Parameters of L1

As with L100, several optionally set parameters are contained within L1 which may be reset by the programmer if he wishes. They are:-

| | |
|---|---|
| A21/L1 | mask for p |
| A22/L1 | mask for q |
| A25/L1 | 'sign' for positive number from accumulator |
| A26/L1 | 'sign' for positive number from B81 |
| A27/L1 | number of characters of floating point exponent |
| A28/L1 } | characters before and after floating point |
| A29/L1 } | exponent. |

Details are given below.

8.11.1 A21/L1, A22/L1

p and q are normally taken modulo 128 and 16 respectively, and this is sufficient for most users. However, it is possible to change these values (which must be powers of 2) by setting suitable masks in A21 and A22. Thus directives

A21/L1 = 255:
A22/L1 = 63

would cause p to be taken modulo 256, and q modulo 64. The maximum values of p and q are 2048 and 512 respectively.

8.11.2 A25/L1

As stated in 8.10.1 and 8.10.2 positive signed numbers are normally preceded by a space in place of a sign. If it is required to output some other character instead, usually a '+' character, A25/L1 should be set to the internal code value of the character required.

Thus,

A25/L1 = K3.5

will produce plus signs before positive numbers, including integers.

8.11.3 A26/L1

In a similar fashion, parameters may be set to the internal value of a character to be printed before positive signed numbers from B81. It is optionally set to K0.1 (SP) within the library routine.

8.11.4 A27/L1

The standard form of the exponent of a floating point number, output by entry A1/L1 has five characters consisting of brackets, a sign and two decimal digits. This may be altered by the programmer to the total number of characters required, say

A27/L1 = 7

If the exponent has less than A27/L1-3 decimal digits, enough spaces will be output after the final bracket to make up the difference. For example with seven characters, an exponent of 3 would be printed

(+3) SP SP SP.

8.11.5 A28/L1 and A29/L1

These parameters are set to the internal values of the characters before and after the exponent for floating point output. If not set by the programmer they will have the values K1.0 and K1.1 respectively to produce '(' and ')'.

Should the programmer wish for other characters in these positions he may reset A28/L1 and A29/L1.

Thus with

A27/L1 = 5
A28/L1 = K1.2 (comma)
and A29/L1 = K0.1 (space)

The floating point number $3.16 \times 10^{17}$ will be output as

SP 3.16,+17 SP

instead of the normal

SP 3.16(+17)

If the last character is to be omitted completely, A29/L1 may be set to zero.





8.12 Input and Output by Extracode

Although sufficient for most purposes, L100 and L1 do not cope with the input requirements of all programs. In particular they cannot deal with paper tapes or cards punched in a non-standard code (i.e. binary input or output). To deal with such cases the input and output extracodes are provided.

These extracodes read six-bit characters from the records of the System Input Tape and write characters to the System Output Tape in preparation for punching or printing. Since information on the Input Tape is in records, provision is made for reading either single characters, groups of characters, or complete records. Output may be formed in similar units.

It is important to note that when L100 is used to read input it takes a complete record from the System Input Tape and reconstructs it before presenting the programmer with a character, a number or a text. For this reason the library routine and the input extracodes can not be used to work on the same input record. Output records may, however, be built partly by L1 and partly by extracodes.

The lack of line reconstruction also makes compound characters impossible with extracode input. Thus a seven-hole tape character may be altered or erased by another character, later in the record. For example, suppose a data tape contained the sequence of characters:

b BS ER a

By reading with L100, the first three characters would be combined to form a compound character and, since it contained erase, this character would be ignored. An entry at A3/L100 to 'read next character' would thus produce the character 'a'. Using an extracode to 'read next character' instead would produce 'b'. Such effects as this must be taken into account by the programmer.

## 8.13 Binary Input and Output

Input and output extracodes may work with either the 12-bit binary characters described in 8.1.4 or 6-bit internal code characters.

On input from punched tape, data documents are read and stored in internal code until one of the markers ***B, ***F or ***E is encountered. These markers are stored. Subsequently a direct binary representation of the input characters is stored as 12-bit characters on the System Input Tape. Any card with a non-standard code punched in its first column will also be stored as binary input. Definitions of the binary markers are given in the chapter on Job Documents.

Seven track input is checked for odd parity when reading binary input and the tape is rejected if an even parity character is encountered. However, if ***P precedes the binary markers ***B, etc., then even parity characters will be accepted and stored as for odd parity characters.

Binary information for five or seven track tape is stored with the three hole side of the punching in the least significant position. Thus the letter M if read in binary from seven track tape would be stored as

000 001 011 . 101

or if read from five track tape, as

000 000 010 . 110

The . indicates the position of the sprocket hole on the tape.

On binary input a punched card is represented by 80 twelve bit characters followed by one six bit zero carriage control character. Each column is stored as a 12-bit binary character with the top bit of the column stored as the most significant bit and the bottom bit as the least significant. Thus one column of a standard ICT punched card is stored as

10 11 0 1 2 3 4 5 6 7 8 9

For output, as described in section 8.4 the least significant bit of the 1060 extracode determines whether output is in internal code or binary.

All binary information, whether input or output is stored as one record with a 0.0 carriage control character. A carriage control character encountered within a section of binary input will be treated as a normal binary character. Thus NL, represented on seven-hole tape by

0000.010

will be stored on the System Input Tape as the 12-bit character

000 000 000 010

Binary input may be used to read tapes or cards punched in non-standard code, since the programmer can provide his own translation routine for the binary representation. Similarly he can cause his results to be punched in a non-standard form by these means.





## 8.14 The Input Extracodes

The Select Extracodes have been described in section 8.4. The remaining input extracodes are defined below. Each is singly modified and refers to the currently selected input.

1052 Find Input Device Number

ba′ = V-store address of the peripheral equipment used for the currently selected input.
If this input originated as output from another program, ba′ = 0.
For input from 5-track tape the least significant bit of ba′ (i.e. bit 23) is 1; otherwise it is 0. The V-store addresses are described in appendix C.

1053 Test Binary / Internal Code

If the next character to be read from the currently selected input stream is a binary character, ba′ = n.
If the next character is in internal code, ba is unaltered.
If there are no characters remaining on the currently selected input stream, an exit is made to the monitor routine.

1054 Read next character to Ba / Jump to n at end of Record

This extracode reads the next 6-bit character from the currently selected input, and places it at the least significant end of ba. With internal code input this will transfer one internal code character. With binary input, where the information is stored in 12-bit characters, the first use of the extracode will read the six most significant bits of the binary character. The next use of the extracode will read the six least significant bits. Normally control will then pass to the next instruction (i.e. b127′ = b127+1) but if the last character, apart from the carriage control character, has previously been read, b127′ = n and Ba contains the carriage control character in bits 18 to 23.
If all characters of the currently selected input stream have been read, this extracode causes an exit to the monitor routine.

1055 ba′ = Number of Blocks Read

This sets in Ba the number of 512 word blocks read from the selected input. In internal code each block holds 4,096 characters, but some of these are carriage control characters and record counts used on the System Input Tape. In binary code, one block holds 2,048 twelve-bit characters.

1056 Read ba characters to S

Before using this extracode the number of 6-bit characters required must be set in the character position of ba. For example, for 18 characters ba must be set to 18D3 or 2.2. The ex-

tracode will then read the next ba characters from the current record of the selected input and place them in Store locations beginning at the half-word address S. Four six-bit characters are packed in each half-word. Bits 22 and 23 of S and bit 0 of ba are ignored.
If the end of the record is not reached, ba is unaltered on exit except for bit 0 which is set equal to one.
If the end of the record is reached no further characters are read and Ba contains the number of characters read in bits 1 to 23. Bit 0 is set to zero. The last character read will be the carriage control character.
If all the characters in the currently selected input stream have already been read, this extracode causes an exit to the monitor routine.

1057 Read next record to S

This extracode reads the next record and places it in the store starting at the half-word address specified in S. Characters will be packed, four six-bit characters to the half-word and bits 22 and 23 of S will be ignored. The last character will be the carriage control character.
On exit Ba contains in bits 1 to 23 the number of 6-bit characters read and bit 0 is zero.
If the record has been partly read, by use of 1054 or 1056, the remaining part of the record is read.
If all records of the currently selected input have been read, this extracode will cause exit to the monitor routine.

Extracodes 1056 and 1057 will run very much faster if no characters have previously been read from the record, or if the number which has been previously read is a multiple of four. Both these extracodes use far fewer instructions per character than does 1054, and are therefore much superior for large amounts of input.

Examples:
1. Input stream 3 consists of one record in internal code followed by a binary marker. Neglect this record and read all the binary information after the marker to store locations 1000.4 onwards, packing the 12-bit characters four to a word. Place the number of 12-bit characters in half-word 1000.

| | | | | |
|---|---|---|---|---|
| 1050 | 0 | 0 | 3 | Select input 3 |
| 1057 | 1 | 0 | 1000.4 | Read internal code record |
| 1057 | 1 | 0 | 1000.4 | Read binary record |
| 124 | 1 | 1 | 0 | Convert character count |
| 124 | 1 | 1 | 0 | To binary character count |
| 113 | 1 | 0 | 1000 | Store character count |

2.a) Read the six-bit characters from input stream five and store them at the bottom of separate half-words beginning at location A5 until a binary record is encountered. All carriage control characters are to be ignored and the address of the last stored character is to be left in B1.





b) Store the first 12-bit binary character at the least significant end of the half-word A6 (assume that at least one such 12-bit character exists).

| | | | | | |
|---|---|---|---|---|---|
| | 121 | 1 | 0 | 0.4 | modifier = 0.4 |
| | 1050 | 0 | 0 | 5 | Select input 5 |
| 1) | 1053 | 127 | 0 | A2 | Go to A2 if next character is binary |
| | 1054 | 60 | 0 | -1* | )Read next ch. to B60 )Check next record if carriage cont. char. |
| | 113 | 60 | 1 | -0.4A5 | Store character |
| | 200 | 127 | 1 | A1 | Go to A1 and add 0.4 to b1 |
| 2) | 124 | 1 | 0 | -1A5 | b1' = last address |
| | 1054 | 60 | 0 | 0 | Read m.s. half of binary character (n is not used) |
| | 125 | 60 | 0 | 0 | Shift b60 up 6 binary places |
| | 113 | 60 | 0 | A6 | Store m.s. half |
| | 1054 | 60 | 0 | 0 | Read l.s. half |
| | 114 | 60 | 0 | A6 | Store l.s. half |

- - - - Next Instruction

3. Read the remainder of the current record of input 2 into locations beginning at 105.4 and place after it, beginning at the next available half word, the same number of characters from the succeeding record. Assume that the succeeding record contains at least as many characters as the current one.

| | | | | |
|---|---|---|---|---|
| 1050 | 0 | 0 | 2 | |
| 1057 | 20 | 0 | 105.4 | |
| 1056 | 20 | 20 | 105.7 | |

## 8.15 The Output Extracodes

The necessary extracodes for selecting output streams have been described in section 8.4. As with input orders each output extracode listed below is singly modified and each refers to the currently selected output.

1062 Find Output Device Type

ba' = V-store address of the peripheral equipment used for the currently selected output.
If this output is to any peripheral (see Job Descriptions, Chapter 10)
then ba' = 0
The V-store addresses are given in appendix C.

1064 Write Character n

This extracode writes the character occupying the six least significant bits of the address to the currently selected output. If the internal code mode has been selected one internal code character will be written. If output is in binary mode, the extracode must be used twice to write the m.s. and l.s. halves of each 12-bit character respectively.

1065 End this Output Record

This writes the carriage control character occupying the six least significant address bits to the currently selected output, and terminates the record. In binary output it is usual to write a zero carriage control character, but in fact the carriage control character is neglected at time of printing or punching and any character would do.

1066 Write ba characters from S

Before entry to this extracode ba must be set as follows:

| | |
|---|---|
| in bits 1 to 23:- | a character count as with 1056 |
| in bit 0:- | 0 if the record is to be ended |
| | 1 if the record is not to be ended |

If the record is to be ended the last character is taken as a carriage control character.
The extracode will then write the ba characters beginning at store address S to the currently selected output. The characters must be packed, four six-bit characters to the half-word. The least significant two bits of S are ignored (i.e. S must be a half-word or full-word address)

1067 Write a Record of ba characters from S

The effect of this extracode is exactly the same as using 1066 with bit 0 of ba equal to zero.
Before entry Ba must contain the character count in bits 1 to 23. Bit 0 of ba will be ignored as will bits 22 to 23 of S.





The extracode will write a record of ba 6-bit characters from store locations beginning at S.
The characters must be packed four to the half-word and the last will be taken as a carriage control character.

Extracodes 1066 and 1067 run very much faster if no characters have previously been sent to the record or if the number of characters previously sent is a multiple of four. Both these extracodes use far fewer instructions per character than 1064 and are to be preferred for large amounts of output.

Examples:

1. Read an internal code record from input one and send it to output three.

| | | | | |
|---|---|---|---|---|
| 1050 | 0 | 0 | 1 | Select input 1 |
| 1060 | 0 | 0 | 3 | Select output 3 |
| 1057 | 1 | 0 | 100 | Read record to locations 100 onward b1' = count of characters |
| 1067 | 1 | 0 | 100 | Output the record |

2. Write the character stored in B2 to output two and follow it by the six characters in store locations 10.4 to 11.1. End the record with the carriage control character in 11.2.

| | | | | |
|---|---|---|---|---|
| 1060 | 0 | 0 | 2 | Select output 2 |
| 1064 | 0 | 2 | 0 | Output ch. in B2 |
| 121 | 21 | 0 | 0.7 | Count of characters in B21 |
| 1066 | 21 | 0 | 10.4 | Output record |

3. Output the following items on stream three, which is a seven-hole tape punch.

a) The characters BI.
b) Three new lines.
c) The 39 twelve bit characters which are stored in packed form, from location A12 onwards.
d) The binary character 0011.010

Then end the binary record.

| | | | | | |
|---|---|---|---|---|---|
| a) | 1060 | 0 | 0 | 3 | Select output 3 for internal characters |
| | 1064 | 0 | 0 | 4.2 | Output B |
| | 1064 | 0 | 0 | 5.1 | Output I |
| b) | 1065 | 0 | 0 | 2.3 | End record with 3N.L's |
| c) | 1060 | 0 | 0 | 3.1 | Select 3 for binary |
| | 121 | 60 | 0 | 39D2 | Set count for 78 six-bit characters |
| | 1066 | 60 | 0 | A12 | Output binary ch's |
| d) | 1064 | 0 | 0 | 0 | Output m.s. half |

| | | | | |
|---|---|---|---|---|
| 1064 | 0 | 0 | K3.2 | Output l.s half |
| 1065 | 0 | 0 | 0 | End binary record |

The following extracodes do not apply to the currently selected output.

1063 Delete Output n

This deletes any information previously sent to Output n, and prevents it being printed, provided it has not been printed already by use of a 1071 extracode (see below).

1070 Rename Output n as Input Ba

This enables information sent to Output n to be read back by the same program as input. For example

1070 3 0 2

will rename output 2 as input 3.

1071 Break Output n

Normally, as described in chapter 10, all output is stored until the running of the program is completed. Then each output stream is put out separately preceded by the heading

OUTPUT n

and the title of the job.
This extracode indicates that the information so far recorded on Output n may, if convenient to the Supervisor, be treated as separate from any subsequent information sent to that output. The Supervisor will then arrange to send all information before the Break to the peripheral and output it with a heading and job title. Subsequent output will be stored in the usual way and output after the job is complete.

1072 Define Output n

Normally output documents should be defined in the job description in the manner given in Chapter 10. They may alternatively be defined by using this extracode. Before obeying this instruction, ba must be set equal to the maximum number of blocks of 4,096 six-bit characters to be allowed on Output n and ba* must define the output device to be used in the code described in appendix C.

Example:
To define a card punch output with number six to which a maximum of 2 blocks will be sent, the following instructions are required:-

| | | | | |
|---|---|---|---|---|
| 121 | 25 | 0 | 2 | Set B25 for 2 blocks |
| 121 | 26 | 0 | J600422 | Set B26 for cards |
| 1072 | 25 | 0 | 6 | Define output 6 |





8.16 Further Information on binary input/output

When using Atlas Internal Code an 80-column punched card is represented by 80 six-bit characters and a next-card carriage control character, i.e. K2.1. Thus, to punch all 80 columns using extracodes 1066 or 1067 it is necessary to specify 81 characters in Ba by an instruction of the type

121 Ba 0 81D3

Similarly to read a card by extracode 1056, ba should specify at least 81 characters.

In binary a punched card is represented by 80 twelve-bit characters, one for each column, and one six-bit character with the value 0.0.

If more than 80 (internal code or binary) characters are output to a card, the first 80 of them will be punched on one card and the remainder on the next card starting again with column one. A continuous stream of characters output to cards with no carriage control information would accordingly be printed 'punched tape fashion' on successive cards.

The punching *** is not recognized on cards. Instead a card is inserted whose first column is punched 7, 8 and whose last column is punched Z, T, B, etc. The intervening 78 columns can contain anything whatsoever.

When a binary tape (but not a deck of cards) is read to its physical end (B or F) the final half word of the stored input is overwritten with J07070707. This means that the last one or two tape characters are stored as 000111000111. The zero carriage control character is unaffected.

On tape following ***E the warning characters ***C or ***Z are themselves stored in binary. They are not subsequently overwritten and each is followed immediately by the zero carriage control character. On cards following 7,8E the terminating card bearing 7,8C or Z is also stored. On tape and cards the new document which follows a C marker is automatically read in Atlas Internal Code; furthermore, if on seven-track tape, the document will be parity checked, even if ***P headed the previous document.

# Chapter 9

## MAGNETIC TAPE

### 9.1 Introduction

Magnetic tape provides an auxiliary store of very large capacity. For many purposes, a magnetic tape can be regarded as a larger but slower form of main store, but it is subject to the restriction that it must be scanned sequentially. It can perhaps best be likened to a notebook whose pages must be turned slowly one at a time: it is possible to ignore a page but it is still necessary to turn it over, and this takes as long as reading it. When using magnetic tape, it is therefore necessary to ensure that the information on the tape is arranged in the order in which it will be required.

Atlas uses two types of magnetic tape, of one and of half inch widths. The system tapes, and most tapes for private use, are one inch wide, pre-addressed tapes, which may be used for fixed or variable length transfers. Reading from the tape is possible in both forward and backward directions. The half inch tape is not preaddressed, and can only be read forwards: transfers are all variable length. One inch tapes prepared on the I.C.T. Orion computer may also be read.

To make efficient use of magnetic tape, it is necessary to overlap magnetic tape transfers and computing as far as possible. This requires care in the timing of transfers and the allocation of storage space when direct transfers to tape are employed. The programmer is, however, relieved of this responsibility when using the extracodes for variable length tape transfers, because these interpose a buffer store between the program and the tape.

Within a program, each magnetic tape is identified by a number. This number, B, is normally written in the Ba digits of an instruction and lies in the range $0 \leq B \leq 99$. The tape number, B, is normally allocated to the appropriate tape by the Job Description, which will be described in Chapter 10.

## 9.2 Atlas One Inch Tape

Information on each magnetic tape is split up into sections of 512 words. There are 5000 sections on each full-length magnetic tape, and these are numbered from 0 at the beginning of the tape to 4999 at the end. Section 0 is reserved for special purposes, and when a tape is first mounted it is positioned ready to move forwards and use section 1. Normally a program will first use the tape starting at section 1. Later it may require to return to section 1 or go on to some other section, and it must then obey a search instruction. The instruction

1001 Ba 0 n

will search for the beginning of section n on tape number Ba prior to fixed length transfers. Thus, to search for section 8 on tape 4, we would write

1001 4 0 8

The 1044 extracode must be used for a search before variable length transfers (see below).

Searching tape is a relatively slow process compared with the computing speed of Atlas, and the time taken is proportional to the number of sections traversed. Therefore the information on tape should normally be stored in consecutive sections starting at section 1, and any search instructions should be given as early as possible in the program.

In this chapter it will sometimes be necessary to refer to "blocks" of store. On Atlas, a block is a unit comprising 512 words of main store; block number P contains the 512 words whose addresses are 512P to 512P + 511. The store structure will be explained in chapter 12 but this simple definition should suffice for the present.

All magnetic tape instructions are singly modified, and throughout this chapter references to the address of an instruction apply to the modified address N + bm. The tape number is normally written in the Ba digits, but if Ba = 122 the tape number is specified by b121.





## 9.3 Block Transfers on One Inch Tape

Block transfer instructions allow a program to transfer 512-word blocks of information between a magnetic tape and a specified block of store. To obtain maximum efficiency in using magnetic tape, a program should use these block transfer instructions and make its own provision for the overlap of tape transfers and computing.

### 9.3.1 Block-Transfer Instructions

The section search instruction, 1001, described in section 9.2, may be used to position the tape before block transfer operations. An instruction of the form

1002 Ba 0 P:

would then read the next 512-word section from tape Ba into block P.

In the block transfer instructions, the octal fraction of the address is used as a parameter K, $0 \leq K \leq 7$, where K + 1 specifies the number of blocks involved in the transfer. Thus, to read the next two sections from tape 4 into blocks 5 and 6, we would write

1002 4 0 5:0.1

The block transfer instructions are as follows:-

| | |
|---|---|
| 1001 | Search for the beginning of section n on tape number Ba. |
| 1002 | Read the next K + 1 sections from tape Ba into store blocks P, P + 1, ..., P + K. |
| 1003 | Read the previous K + 1 sections from tape Ba into store blocks P + K, ..., P + 1, P. |
| 1004 | Write store blocks P, P + 1, ..., P + K on to the next K + 1 sections of tape Ba. |
| 1005 | Move tape Ba forwards K + 1 sections. |
| 1006 | Move tape Ba backwards K + 1 sections. |

When reading either forward or backward, information will be held in store in the same order as on tape, with the first word in the lowest numbered tape section transferred to the start of the store block with the smallest address. This order of words is also maintained when writing to tape.

Examples:

1. Read section 19 of tape 3 to main store block 6.

| | | | | | |
|---|---|---|---|---|---|
| (a) | 1001 | 3 | 0 | 19 | Search for section 19 |
| | 1002 | 3 | 0 | 6: | Read forward to block 6 |
| (b) | 1001 | 3 | 0 | 20 | Search for section 20 |
| | 1003 | 3 | 0 | 6: | Read backward to block 6 |

The information in block 6 will be the same in either case.

2. Read sections 1, 3, 5, 7 and 9 of tape 66 into main store blocks 20 to 24 inclusive. The previous operation on tape 66 was to write to section 13.

| | | | | | |
|---|---|---|---|---|---|
| | 1006 | 66 | 0 | 0.2 | Position tape after section 11 |
| | 121 | 1 | 0 | 4: | Set block modifier |
| 1) | 1006 | 66 | 0 | 0 | Move tape back 1 section |
| | 1003 | 66 | 1 | 20: | Read previous section |
| | 123 | 1 | 0 | 1: | Reduce block modifier |
| | 216 | 127 | 1 | A1 | Return if non-negative |

A considerable saving is obtained in this example by reading backwards. To have searched for section 1 and read forwards would have meant traversing nine sections twice and would have taken almost twice as long.

The instructions have been arranged so that the 1006 instruction comes before the 1003 instruction in the loop. If the 1003 instruction had been put first, the program would have traversed one extra section after the last read instruction; in this particular program the extra section would have been section 0 and the program would have been monitored because the use of section 0 is prohibited.

9.3.2 Use of Block Transfers.

The way in which a program uses magnetic tape will depend very much on the requirements of the process it is performing. Sometimes it is necessary to read a large amount of information, such as a complete matrix, before computing can commence. In this case, shortage of store may prevent the overlap of computing with further tape reading, but at least the next required tape address can be searched for; afterwards it may be possible to overlap the writing of the results to one tape with the reading of the next set of data from another. The technique of branching, to be described in chapter 12, may also help in this situation.

When it is possible to work sequentially through the information on tape, operating on one word or one small group of words at a time, considerable savings can be made by overlapping tape transfers with computing. This is done automatically by the variable length transfers (see below). With block transfers, overlap can be obtained by transferring alternately to two different blocks, computing on one whilst transferring to the other.

The same process can be used when operating on two or more magnetic tapes. When processing longer items, special care is needed if an item overlaps two tape sections.

Example:

To read sections 1 to 2000 of tape 4, presenting each word to a processing routine R3, a control program of the following form would suffice:





| | | | | | |
|---|---|---|---|---|---|
| | R0 | ASSUMED | | | |
| | 1001 | 4 | 0 | 1 | Search for section 1 |
| | 1002 | 4 | 0 | 5:0.1 | Read to blocks 5 and 6 |
| | 121 | 71 | 0 | 511 | Set word count |
| | 121 | 72 | 0 | 1999 | Set section count |
| | 121 | 127 | 0 | A4 | Jump to label 4 |
| 2) | 124 | 2 | 0 | 1 | Step up address |
| | 203 | 127 | 71 | A1/3 | Count words in block |
| | 121 | 71 | 0 | 511 | Reset word count |
| | 203 | 127 | 72 | *2 | Count tape section |
| | 121 | 127 | 0 | A5 | Exit |
| | 1002 | 4 | 2 | -1: | Read to refill block just emptied |
| | 203 | 127 | 73 | A1/3 | Count blocks |
| 4) | 121 | 73 | 0 | 1 | Set block count |
| | 121 | 2 | 0 | 5: | Set first block address |
| | R3 | | | | |
| 1) | 324 | 0 | 2 | 0 | Read word |
| | : | : | : | : | Process word |
| | 121 | 127 | 0 | A2/0 | Return to R0 for next word |

In the preceding example, because it is reading two sections in advance, the program reads one section more than it requires, but does not attempt to process the extra section. This extra read operation could be avoided by an extra test in the example, but it would be unavoidable if the end of the process were detected by the processing routine on receipt of the last word. There is normally no harm in reading extra sections provided that they do not lie outside the range 1 to 4999 inclusive, and provided that these extra sections have been written to since the tape was last addressed. It is therefore advisable to write a few extra sections after the information when a tape is written to. A magnetic tape fault (512-word fault) will show up if an attempt is made to read from a section not previously written to.

## 9.4 Variable Length Working on One Inch Tape

To simplify the writing of some magnetic tape programs, extracodes are provided which execute the transfer of variable length records between magnetic tape and the main store. Variable length operations must initially be preceded by the variable length word search. The instruction

```
1044    Ba    0    S
```

will search tape number Ba for the section and word contained in the full word with address S. The section number is contained in the more significant, and the word number in the less significant half word, both being held as 21 bit integers. Thus, to search for the eighth word of section 10 on tape 4, we would write

```
1) H10    8
   1044   4    0    A1
```

The 1001 search may not be used with variable length working.

The extracodes for variable length operations require an area of store to be used as a "buffer", to hold information in transit between the tape and the store. This buffer must be set up, and the mode of operation specified, by obeying a "start" extracode for each tape involved in variable length operations. Thereafter, a "transfer" extracode is used to transfer information between the buffer and the program as required. When writing to tape each such transfer forms one "record" on the tape.

Before writing variable length records to magnetic tape, it is necessary to obey a 1032 instruction. This "start writing" instruction normally takes the form:

```
1032    Ba    0    P:0.K
```

This prepares for writing forwards starting at the next word on tape Ba, and selects it for variable length operations. It also sets up a buffer store in blocks P to P + K inclusive; normally K = 1, allowing a two block buffer. Thus, to start writing variable length records to tape 5, using main store blocks 10 and 11 (locations 5120 to 6143) as buffer, we would write

```
1032    5    0    10:0.1
```

Thereafter, information may be transferred to tape 5 by 1040 instructions. Before obeying a 1040 instruction, when writing to tape, the number of words to be transferred and the end-of-record marker must be set in an index register: the number in the integral part and the marker in the octal fraction. This index register must then be specified in the Ba digits of the 1040 instruction. Normally, the end of an ordinary record should have a marker of value 1. Thus, using B6 to specify a transfer of 25 words, we would write

```
121    6    0    25.1
```

To transfer 25 words (as specified in B6) starting at address 2000, we would then write

```
1040    6    0    2000
```





Example:

Given a 30 x 100 matrix stored by rows in location 8000 onwards. Write the 30 rows, of 100 numbers each, as 30 separate records starting at the beginning of section 8 on tape 4.

```
1)H  8     0
     1044  4    0   A1       Search for section 8
     1032  4    0   10:0.1   Start Writing to tape 4
     121   1    0   29       Set row count
     121   2    0   0        Clear modifier
     121   3    0   100.1    Prepare to transfer 100 words
5)   1040  3    2   8000     Transfer
     124   2    0   100      Increase modifier
     203   127  1   A5       Count rows
```

Before reading variable length records from tape, it is necessary to obey a "start reading" instruction. To start reading forwards, a 1030 instruction must be used, and this normally takes the form

```
1030    Ba    0    P:0.K
```

This starts reading forwards from the next word on tape Ba and selects it for succeeding variable length operations. It also sets up a buffer in blocks P to P + K inclusive; normally K = 1, giving a two block buffer. Thus, to start reading variable length records from tape 5, using main store blocks 10 and 11 as buffer, we would write

```
1030    5    0    10:0.1
```

Thereafter, information may be transferred from tape 5, reading forwards, by using 1040 instructions. The Ba digits of the 1040 instruction indicate which index register has been used to specify the amount of information to be transferred. When reading from tape, this index register normally specifies the maximum number of words to be transferred. It may also specify an end-of-record marker whose purpose will be explained later. The number is specified by the integral part, and the marker, if required, by the octal fraction.

To read one record at a time, the maximum length of record should be specified and the marker should be zero (or one). After the transfer, the same index register records in its integral part the number of words actually read, and in its octal fraction the value of the marker at the end of the record. Thus, to read a record of not more than 200 words to location 1500 onwards, we would write

```
121     10    0    200
1040    10    0    1500
```

If the actual record were of 100 words terminated by a marker 1, then B10 would contain 100.1 after the transfer.

Thus, to read to location 200 the first row of the matrix recorded on section 8 in our previous example, we would write:

| | | | | | |
|---|---|---|---|---|---|
| 8)H | 8 | 0 | | | |
| | 1044 | 4 | 0 | A8 | Search for section 8 |
| | 1030 | 4 | 0 | 10:0.1 | Start reading from tape 4 |
| | 121 | 3 | 0 | 100 | Prepare to read up to 100 words of next record |
| | 1040 | 3 | 0 | 200 | Transfer |

When reading, it is possible to ignore end-of-record markers of less than a given value by specifying that value in the octal fraction of Ba before the transfer. Thus, for example, by setting 300.2 in B3 before the 1040 instruction above, we could read the first 300 elements of the matrix; the marker 1 written at the end of each row is less than the octal fraction 2 set in B3 and would therefore be ignored.

The instructions to start reading or writing assume the tape to be positioned at a marker, either by previous variable length operations or word search.

So far we have only considered working on one magnetic tape at a time, and in this case the start instruction selects that tape for all succeeding tape operations. When working on two or more tapes, it is still necessary to use start instructions to initiate variable length working, but subsequently 1033 instructions must be obeyed to select whichever tape is required. This "select" instruction chooses tape number Ba for succeeding variable length transfers until the next select or start instruction.

Example:

Tape number 2 contains a file of variable-length records starting at section 1. Each record is terminated by a marker 1, except the last record which is terminated by a marker 2. The maximum length of record is 50 words. Copy the file to tape 3, section 1 onwards.

| | | | | | |
|---|---|---|---|---|---|
| 2)H | 1 | 0 | | | |
| | 1044 | 2 | 0 | A2 | Search for section 1, tape 2 |
| | 1044 | 3 | 0 | A2 | Search for section 1, tape 3 |
| | 1030 | 2 | 0 | 10:0.1 | Start reading, tape 2 |
| | 1032 | 3 | 0 | 12:0.1 | Start writing, tape 3 |
| 1) | 1033 | 2 | 0 | 0 | Select tape 2 |
| | 121 | 4 | 0 | 50 | Read next record to location A6 onwards |
| | 1040 | 4 | 0 | A6 | |
| | 1033 | 3 | 0 | 0 | Select tape 3 |
| | 1040 | 4 | 0 | A6 | Write record |
| | 210 | 127 | 4 | A1 | Jump if marker odd, End if marker even. |





9.4.1 Variable Length Instructions

In the previous section, a selection of the most important variable length magnetic tape instructions have been described and illustrated in order to explain how they are used. In this section, the full range of variable length tape instructions will be defined, and there will be some repetition of information from the previous section.

When using variable length tape transfers, the information is stored on tape in groups of words known as "records", with a 21-bit count and a 3-bit marker on each side to denote the ends of the record. Thus the space on tape occupied by a record is one word more than the number of words of information. Each writing transfer forms one record on tape. A reading transfer may either read a specified number of words or read up to the end of a record; in both cases markers are omitted from the transfer. Instructions to start reading or writing must only be given when the tape is positioned at a marker.

A number of consecutive records often form a larger unit, such as a complete matrix or a complete file, and it is often desirable to mark this in some way. For this reason eight orders of marker, numbered 0 to 7, are provided. Ordinary records may be terminated by a marker of order 1, groups of records by a marker of order 2, and so on up to 7, which normally denotes the beginning or end of a file. Reading transfers may then read up to a marker of a specified order, and the program may test the value of the marker read. Use of the 0 marker is not recommended.

Variable length working must always be initiated by a start instruction. This sets up a buffer store, selects the tape to be operated upon, and specifies the mode of operations, whether write, read forwards or read backwards. A separate start instruction must be given for each tape on which variable length transfers are required, but thereafter a select instruction may be used to choose the tape to be used. The transfer instruction operates on the tape which was last selected by a start or select instruction, and transfers information in the mode selected for that tape. To change the mode it is necessary to obey another start instruction. Unless the tape is already in variable length mode, the instruction to start writing begins by writing a marker of order 7. To start writing without commencing with a 7 marker, the sequence Word Search, Start Reading Forwards, Start Writing, Transfer should be used. Alternatively, the 1042 'mark' instruction may be used after the instruction to start writing.

A start instruction always initiates variable length transfers to or from the next word on tape or the previous word in the case of reading backwards. To begin working at a particular address on tape, the start instruction must be preceded by the word search instruction, 1044; when starting to read variable length records, this starting address must be the address of a marker at one end of a record.

The variable length writing operations do not provide a means of overwriting selected words on a magnetic tape: complete new blocks are formed in the buffer and the previous contents of the tape are not preserved. If it is required to preserve the beginning of the first section, this may be done by preceding the start writing instruction by a start reading forwards instruction with K = 0.

It is important to note that, when using a buffer of K+1 blocks, the attempt to read variable length records backwards from any of the first K sections, or forwards from any of the last K sections, will lead to an end-of-tape interrupt. K sections should therefore be left unused at both ends of the tape.

### 9.4.2 Start and Select Instructions

In the following specifications, P is a block address, and K an octal fraction, as above.

1030 Start Reading Forwards

Start reading forwards from the next word on tape Ba and select it for variable length operations. Set up a buffer in blocks P, P+1,..., P+K.
If this instruction is given when the tape is already in read forwards mode, then the tape will be wrongly positioned.

1031 Start Reading Backwards

Start reading backwards from the previous word on tape Ba and select it for variable length operations. Set up a buffer in blocks, P, P + 1, ..., P + K.

1032 Start Writing Forwards

Prepare to write forwards starting at the next word on tape Ba, and select it for variable length operations. Set up a buffer in blocks P, P + 1, ..., P + K.
Also, write a marker 7 before the first word of information, provided that the given tape is not already in use for variable length working. If the tape is already in use for variable length working, the marker written will be equal to the marker at the end of the previous record.

The three 'start' instructions 1030, 1031 and 1032 must be preceded by a 1044 word search, even when the desired word is the first in the section. To give predictable results, the tape must be positioned at a marker whenever a 'start' instruction is used. The K+1 block buffer must have been allowed for in the job description (see Chapter 10). If one of these blocks is already in use by the program, the information in it will be lost. Strings of information with markers at either end are transferred to and from these blocks by the 1040 instruction. When a buffer block has been filled in the writing mode, it is transferred to tape; when completely read in the reading mode, the next section is read from tape. The buffer blocks are not protected from the program, but should not be referred to directly.

It may be desirable to read variable length records from a tape for a while, and then begin writing to the very next record. 1032 will switch modes in this way; the first record written will begin with the same marker as that terminating the previous record.

1033 Select

Select tape Ba for succeeding variable length operations, in the mode specified by the preceding start instruction for that tape. All succeeding 1040, 1041 and 1042 instructions apply





to tape Ba until another tape is selected. If a write buffer had previously been set up for tape Ba, its contents will be written to tape, and a new buffer set up. Records will be written as if tape Ba were selected throughout.

Extracodes 1030 and 1031 above assume that the tape to be read from has been previously written by variable length operations. If this is not the case, the extracodes 1034 and 1035 should be used instead.

1034 Start Reading Forwards From Fixed Blocks

As 1030, but operating on a tape which has not been written in the form of variable length records.

1035 Start Reading Backwards From Fixed Blocks

As 1031, but operating on a tape which has not been written in the form of variable length records.

### 9.4.3 Transfer and Organizational Instructions

Notation: $b_w$ = Integral part of ba (bits 1 to 20) $(0 \le b_w < 2^{19})$

$b_k$ = Octal fraction of ba. $(0 \le b_k \le 7)$

1040 Transfer

Transfer up to $b_w$ words between store addresses starting at S and the selected tape, in the mode (reading forwards, reading backwards, or writing) appropriate to that tape. On writing, $b_w$ words from locations S, S + 1, ..., S + $b_w$ - 1 are written to the next $b_w$ locations on the selected tape. A marker $b_k$ is written on tape after them. On Reading, provided that $b_w \neq 0$, the transfer continues until $b_w$ words of information have been read or until a marker $\ge b_k$ is encountered, whichever is the sooner.

$b_w'$ = the number of words of information actually read.

$b_k'$ = 0 if no marker $\ge b_k$ was encountered.

= m if a marker m $(\ge b_k)$ terminated the transfer or immediately followed the last word transferred.

When reading forwards, the next $b_w'$ words are read from tape to store locations

$$S, S + 1, \ldots\ldots, S + b_w' - 1$$

When reading backwards, the previous $b_w'$ words are read from tape to store locations

$$S, S - 1, \ldots\ldots, S - b_w' + 1$$

If $b_w = 0$ when reading forwards, the transfer continues until the first marker $\ge b_k$ is encountered. When reading backwards with $b_k$ and $b_w$ zero, the transfer continues until the end of the first record, and the $b_w'$ words of the record are read to store locations

$$S,\ S-1,\ \ldots\ldots,\ S-b_w'+1$$

It is not advisable to write with the octal fraction $b_k = 0$, because when the resulting string, which therefore ends with a zero marker, is read back, the octal fraction $b_k$ will be zero regardless of whether reading ended at the zero marker or somewhere short of the marker within the string itself.

1041 Skip

Skip $b_w$ words, terminating on a marker $b_k$.

Skip operates in the same way as transfer, except that no words are transferred to or from the program store.
When in a writing mode, $b_w$ addresses on tape are skipped and a marker $b_k$ is written after them. Note, however, that the previous contents of these addresses, whether information or marker, are not preserved on tape, except when complete 512-word tape sections are skipped.
When in a reading mode, the skip continues until $b_w$ words of information have been passed or until a marker $\geq b_k$ is encountered, whichever is the sooner.
$b_w'$ = the number of words of information actually skipped.
$b_k' = m$ if a marker m ($\geq b_k$) terminated the transfer or came immediately after word b.

Note that skip is less efficient than search for moving long distances along the tape, and should not be used for skipping more than a few sections.

1042 Mark

Available only when in writing mode.
Writes a marker K ($0 \leq K \leq 7$) after the last word on the selected tape. This marker replaces any marker which was previously on the tape at this point.
After writing a string on tape, it may be discovered that the end of a group has been reached. The mark instruction may then be used to change the marker at the end of the string.
It may be used again if it is later found that the end of an even higher order group has been reached.
A mark instruction may also be used immediately after a start-writing instruction, to write the specified marker before the first record to be written.

1043 Stop Variable Length

Stop variable length operations on tape Ba.
After variable length operations for a given tape have been completed, a stop instruction may be given; it will release the buffer blocks associated with those operations. After





writing operations, it will cause the last part-section to be written from the buffer to magnetic tape. The following instructions also have the effect of stopping previous variable length operations:

start, search, unload, release tape, end program.

Whenever variable length writing is terminated on a given tape each buffer block that contains information transferred there by a 1040 instruction is written to tape. A buffer block is not written to tape unless it contains such information. This means that the first few words of the last buffer block written may contain the end of the final record, or even just the final marker, and the rest of the block contain perhaps the remains of some previously transferred records. This fact permits one to overwrite individual records without disturbing the records on either side. This is done by filling the buffer with the record that is to be overwritten along with the records, or part of the records, on either side of it. One aligns on the marker at the beginning of the record and starts reading forward (1030) with a buffer large enough to contain the whole record at one time. One then skips (1041) the record in question, starts reading backward (1031, with the same buffer) and skips back over the record to the beginning marker. One switches to write mode (1032, same buffer again) and writes the new string in its proper place. A 'stop' instruction (1043) will then finish the job by causing the buffer blocks, now containing the new string in place of the old, to be written back to the tape.
The two uses of the skip instruction are necessary to make the marker at the beginning of the old string available for the construction of the marker at the beginning of the new string.

1044 Word Search

Search tape Ba for the section and word specified in the full word with address S. Stop variable length operations on tape Ba. The section and word are given as 21 bit integers in the more and less significant halfwords in full word s respectively.
The section number must be greater than 0, and less than 5000; the word number is taken modulo 512. This instruction must be used to align the tape on a marker before using a 'start' instruction; after the word search, a 'start' instruction is necessary before further variable length operations on tape Ba.

1036 Set ba' = number of selected tape

Place in bits 10-16 of Ba the program number of the tape currently selected for variable length operation. If there is no tape currently selected, ba' will be negative. Bits 10-16 correspond to the Ba position in the more significant halfword of an instruction. One application is to enable a sub-routine to select a different tape for variable length transfers, and then to re-select the original tape before the main program is re-entered.

1037 Store 'mode of tape Ba' in S

The transfer mode at present selected for tape Ba will be indictated by placing in half-word S the appropriate integer from the following table:-

0 Variable length transfers, reading forwards from variable length records

1 Variable length transfers, reading backwards from variable length records

2 Variable length transfers, writing variable length records.

3 Not currently selected for variable length transfers.

4 Variable length transfers, reading forwards from fixed length blocks.

5 Variable length transfers, reading backwards from fixed length blocks.

Examples:

1. Tape 1 contains a file of variable length records and tape 2 contains amendments to this file. The information on each tape starts at section 1 and the records are not more than 40 words long. Each record is terminated by a marker 1, except the last record which is terminated by a marker 2. Each record is identified by a key in its first half-word, and the records in each file are sorted in ascending order of keys. It is required to form an updated file on tape 3 by inserting the amendment records in place of the corresponding records on the original file.

| | | | | | |
|---|---|---|---|---|---|
| 1)H | 1 | 0 | | | |
| | 1044 | 1 | 0 | A1 | Initiate variable length operations on tapes, 1, 2 and 3 |
| | 1030 | 1 | 0 | 20:0.1 | |
| | 1044 | 2 | 0 | A1 | |
| | 1030 | 2 | 0 | 22:0.1 | |
| | 1044 | 3 | 0 | A1 | |
| | 1032 | 3 | 0 | 24:0.1 | |
| 4) | 1033 | 2 | 0 | 0 | Read Amendment Record |
| | 121 | 12 | 0 | 50 | |
| | 1040 | 12 | 0 | 100 | |
| 5) | 1033 | 1 | 0 | 0 | Read Main-File Record |
| | 121 | 11 | 0 | 50 | |
| | 1040 | 11 | 0 | 150 | |
| | 1033 | 3 | 0 | 0 | Select Updated File |
| | 101 | 10 | 0 | 150 | Go to A7 if Amendment Key equals Main-File Key |
| | 102 | 10 | 0 | 100 | |
| | 214 | 127 | 10 | A7 | |

(Program continues on next page)





| | | | | | |
|---|---|---|---|---|---|
| | 1040 | 11 | 0 | 150 | Write Main-File Record |
| | 210 | 127 | 11 | A5 | Go to read next record if marker is odd |
| | 1117 | 0 | 0 | 0 | End Program |
| 7) | 1040 | 12 | 0 | 100 | Write Amendment |
| | 210 | 127 | 12 | A4 | Go to read next amendment if amendment marker is odd |
| | 1042 | 0 | 0 | 0.1 | Write a marker one |
| | 210 | 127 | 11 | A5 | Go to read next Main-File record if marker is odd |
| | 1042 | 0 | 0 | 0.2 | Mark end of updated file |
| | 1117 | 0 | 0 | 0 | End program |

2. There is a 50 word record in section 10 of tape 33 which is preceded by a '7' marker in word 177. Replace that record with the 50 consecutive words beginning in the store at A3, leaving the '7' marker undisturbed.

| | | | | | |
|---|---|---|---|---|---|
| 3) | H10,177 | | | | Section 10, word 177 |
| | 1044 | 33 | 0 | A3 | Search for marker |
| | 1030 | 33 | 0 | 1: | Fill the one block buffer |
| | 1041 | 0 | 0 | 0 | Skip to the end of next record |
| | 1031 | 33 | 0 | 1: | Read backward |
| | 1041 | 0 | 0 | 0 | Skip back to head of record |
| | 1032 | 33 | 0 | 1: | Switch to write mode |
| | 121 | 75 | 0 | 50 | |
| | 1040 | 75 | 0 | A3 | Replace record in buffer |
| | 1043 | 33 | 0 | 0 | Write block 1: to section 10 |

Although the length of the string was known in advance and B0 mentioned in both of the skip instructions, one could use one of the skip instructions to pick up the length of the string in a B-line.

If the 1032 write instruction is replaced by a 1042 mark instruction, the program will replace only the mark at the head of the string.

### 9.4.4 Efficiency of Variable Length Working

The basic magnetic tape operations on Atlas transfer information in blocks of 512 words, but the variable length instructions disguise this fixed block structure. Provided that the length of transfers is small compared with the size of the buffer, these instructions also provide overlap of tape transfers and computing. To achieve these effects, extra words are written on the tape as markers and extra instructions are obeyed to transfer the information between the buffer and the program store. Provided that the variable length records are not too short, the loss of efficiency is not high. With a two-block buffer and records of 50 to 150 words, the use of variable length instructions might be expected to increase the cost of a job by perhaps 1% or 2%. This efficiency will be maintained with longer transfers also, but the automatic overlap of transfers with computing will be lost as the transfer size approaches the buffer size.