INTERNATIONAL COMPUTERS AND TABULATORS LIMITED

Input And Output
For
I.C.T. Atlas Computers 1 & 2

This Document is a corrected and expanded version of Chapter 8 of the Atlas Provisional Programming Manual (CS 348) and supersedes the Provisional corrected version of Chapter 8 dated September 1963.

APRIL 1964

I.C.T. Limited,
London Computer Centre,
68 Newman Street,
London, W.1.

I.C.T. Limited,
Television House,
Peter Street,
Manchester, 2.

CONTENTS

Contents (Continued)

This chapter explains the use of library routines L1 and L100 and the input and output extracodes. It also shows the form in which input data must be prepared and explains the internal code used by Atlas as well as the handling of 'pure' binary information.

1.1 Introduction

1.1.1 Peripheral devices. Atlas can control a large number of input and output mechanisms and these peripheral devices are of various forms. Five- and seven-hole paper tape and punched card readers are used for input, while output may be produced on paper tape, cards or a line printer.

The working rate for such peripherals is of the order of 10^3 characters per second. This is very slow compared with the processing rate of the central computer which is of the order of 2×10^6 characters per second. Clearly, conventional methods of input and output, in which each character is read from a peripheral as required or written away as soon as it is produced, would lead to considerable inefficiency in Atlas. For this reason, a 'buffering' system is adopted.

1.1.2 The System Input and Output Tapes. All the information required for a program is fed through the slow peripherals and is automatically stored in a standard form on a magnetic tape. This tape is called the System Input Tape. Work on the program will not begin until all the required information is stored in this way. Similarly the output from a program is stored on magnetic tape (the System Output Tape) and transferred to the peripheral only after the program is completed. In this way the central computer can be occupied with other work while the actual transfer to and from the peripheral takes place.

1.1.3 Internal Code Input and Output. Normally during the transfer from peripheral to the System Input Tape each character (i.e. a paper tape character or card column) is automatically translated into a six-bit character in Atlas Internal Code. Similarly each character for output is represented in this code on the System Output Tape. The six-bit characters are packed 8 to a word and stored in 'records' (see 1.3 below).

Seven-track input is checked for odd parity and rejected on parity failure. Rejection is prevented by preceding the seven track information by ***P in which case an even parity character is replaced by the fault character 77 (six binary ones) and reading continues. There is no parity check when reading 5-track tape. If the first column of a card is a non-standard punching the card is stored in binary (see 1.12); if the first column is standard but subsequent columns are non-standard they are represented by the fault character 77.

Equivalent characters on punched cards, 5- or seven-hole tape and the line printer, are represented by the same character in internal code. In this way one input routine can deal with input from any peripheral and one output routine can build up output for any output mechanism.

Thus, for example, the character 'M', whether it has been read from five- or seven-hole tape or from cards, will be represented on the System Input Tape as

K5.5 or, in binary 101101 .

Similarly if 'M' is to be output it will be stored on the System Output Tape in the same way, regardless of the output peripheral for which it is intended. Further details of the internal code are given in 1.2.

Input and output in internal code may be carried out by either the input and output library routines L100 and L1 or by input and output extracodes. Both forms are described in detail below.

1.1.4 Binary Input and Output. The programmer can, if he wishes cause a direct binary representation of the holes on the paper tape or cards to be stored on the System Input Tape, instead of the internal code representation. This is known as binary input and must be read by extracode and not the library routine L100. Each character so stored will occupy 12 bits and characters will be packed four to a word.

For example, 'M' is represented on seven-hole paper tape by the character

1011.101

where ones represent holes and the full stop represents a sprocket hole. As binary input this would be stored on the System Input Tape as

000 001 011 101

On the other hand the character 'M' is represented on a punched card by holes in the '-' and '4' positions. As binary input this is stored as the 12 bit character

010 000 100 000

(see 1.12 and 1.15)

In a similar fashion, by the use of extracodes, a twelve bit character may be output in binary form. It will be stored on the System Output tape in this form and, when sent to the peripheral, holes will be punched in positions corresponding to the one bits.

Binary input and output are particularly useful when dealing with non-standard paper tape or card codes. More details are given in 1.12 and 1.15.

1.2 The Internal Code.

Each character is represented in internal code by six bits. When a single six-bit character is held in a 24-bit index register it is usually stored in the least significant six bits. It is thus represented by two octal integers separated by a point, and this notation will be used throughout this chapter.

The use of six bits imposes an upper limit of 2^6 or 64 distinct characters. To extend this number, two sets of characters are introduced, the inner set and the outer set. Most of the commonly used characters are contained in the inner set. A character in the outer set will have the same six-bit representation as an inner set character but they will be distinguished in the following way.

Every line, whether input or output, begins in inner set and all characters are taken to be inner set until a 'shift to outer set' character is encountered. All characters will then be interpreted as outer set members until a 'shift to inner set' character or the end of the line is reached.

On output, an internal code character which can not be represented by a character on the required peripheral will be replaced by a full stop on the Anelex printer or the card punch, and by erase on 5 or 7 track tape. Thus an attempt to print ≥ on an Anelex printer would result in a full stop being printed. This also applies to non-printing characters such as back space and tabulate which are only available on seven-hole tape.

1.2.1 Abbreviations. In the Internal Code Table of 1.2.2 the following abbreviations are used:-

BS	=	Back Space	SI	=	Shift to Inner set
ER	=	ERase	SO	=	Shift to Outer set
FS	=	Figure Shift	SP	=	SPace
FT	=	Fault	TB	=	TaBulate
LC	=	Lower Case	UC	=	Upper Case
LS	=	Letter Shift	UL	=	UnderLine

The availability of characters on the different peripheral devices is indicated as follows:-

u	All peripheral devices
a	Anelex Line Printer
7	Seven-hole paper tape
c	Punched cards
5	Five-hole paper tape

Characters in parentheses are alternatives available on commercial seven-hole or five-hole paper tape codes, as used on Orion and Pegasus.

1.2.2 The Internal Code Table

Internal Code	Character				Internal Code	Character			
	Inner Set		Outer Set			Inner Set		Outer Set	
0.0					4.0	'(n)	u		
0.1	SP			u	4.1	A	u	a	7
0.2	TB	7	%	a	4.2	B	u	b	7
0.3	BS	7	£	a	4.3	C	u	c	7
0.4	SO				4.4	D	u	d	7
0.5	SI				4.5	E	u	e	7
0.6	LC/LS			7/5	4.6	F	u	f	7
0.7	UC/FS			7/5	4.7	G	u	g	7
1.0	(	u			5.0	H	u	h	7
1.1	)	u			5.1	I	u	i	7
1.2	,	u			5.2	J	u	j	7
1.3	π (£)	u			5.3	K	u	k	7
1.4	?	u			5.4	L	u	l	7
1.5	&	a7			5.5	M	u	m	7
1.6	*	u			5.6	N	u	n	7
1.7	/	u	:	a7c	5.7	O	u	o	7
2.0	0	u	ø(X)	5	6.0	P	u	p	7
2.1	1	u	[	a7c	6.1	Q	u	q	7
2.2	2	u	]	a7c	6.2	R	u	r	7
2.3	3	u	→	5	6.3	S	u	s	7
2.4	4	u	≥	5	6.4	T	u	t	7
2.5	5	u	≠	5	6.5	U	u	u	7
2.6	6	u	UL	a7c	6.6	V	u	v	7
2.7	7	u	|	a7c	6.7	W	u	w	7
3.0	8	u	2 (%)	a7	7.0	X	u	x	7
3.1	9	u	≃ (v)	5	7.1	Y	u	y	7
3.2	<	a7c	& (10)	a7	7.2	Z	u	z	7
3.3	>	u	β (11)	a7	7.3				
3.4	=	u	½	a7	7.4				
3.5	+	u	10	a	7.5				
3.6	-	u	11	a	7.6				
3.7	.	u			7.7	FT	7c	ER	75

1.2.3 Shifts and Case Changes. UC, LC, FS and LS are not stored on input when they are used to change from one case or shift to another. Each of these characters merely alters the meaning of the characters which follow on the paper tape and this alteration will be allowed for when these characters are translated into internal code.

Thus the sequence

Tape (7)

on seven-hole paper tape would be punched as

T LC a p e SP (UC 7 LC)

but it would be stored on the system input tape as

T SO a p e SP SI (7)

Redundant shift characters, such as FS when already in figure shift are stored, however, since they may indicate a fault.

Similarly UC, LC, FS and LS need not be used when preparing output. The internal code representation of the character is specified and shift and case changes will be automatically inserted where necessary.

For example outputting the internal code characters

W 2

to five-hole paper tape will cause the characters

LS W FS 2

to be punched.

1.3 Carriage Control Characters and Records

As mentioned in 1.1.3 input is stored on the System Input Tape in records. These records correspond to one line of printing on paper tape or one card if the input is on cards. The last character of a record is called a carriage control character and is not represented in internal code. Carriage control characters have a special code of their own. Input from paper tape or punched cards can give rise to only the following carriage control characters:-

Code	Character
2.1	Newline (7-hole tape) or End-of-card
4.0	Paper Throw (7-hole tape)
2.0	Carriage Return (5-hole tape)
0.1	Line Feed (5-hole tape)

These are abbreviated to NL, PT, CR and LF respectively.

On output, records are built up on the System Output Tape. The last character of each record is interpreted as a carriage control character as follows:-

Codes	Effect
0.0 to 1.7	0 to 15 line feeds without carriage return
2.0 to 3.7	0 to 15 line feeds with carriage return
4.0 to 4.7	Paper throw on channels 0 to 7 without CR
5.0 to 5.7	Paper throw on channels 0 to 7 with CR

Carriage control facilities, and hence the interpretation of carriage control characters, vary from one output device to another.

The number of line feeds is always performed correctly but the following restrictions apply to other facilities.

On the Anelex Line Printer

Line feed and Paper Throw are always accompanied by carriage return.

Channel 0 is the 'top of form' channel. Thus 4.0 and 5.0 mean carriage return to the top of the next form. Channels 1 through 7 provide vertical spacing (always with carriagee return) as determined by a loop of paper tape which is fitted to the paper throwing mechanism of the Anelex printer.

The width of a line printed on the Anelex is 120 characters. If more than 120 characters are output to a line a new line is automatically begun and the excess characters are printed on it.

On the Card Punch

Line feed means 'next card'.
Paper throw is replaced by one 'next card' (i.e. 2.1)
Carriage return is ignored.

On Seven-hole Tape

Line feed is always accompanied by carriage return.
Carriage return without line feed (i.e. 2.0) is ignored.
Paper throw is never accompanied by carriage return. The paper throw character on tape will only take effect if the flexowriter on which it is printed has a paper throw facility. Otherwise one 'new line' is output.

On Five-hole Tape

Paper throw is replaced by one line feed.

The channel number for paper throw will be taken modulo m, where m is the number of homing channels available on the printer.

The carriage control character 0.0 is ignored by each of the four types of equipment. The character 2.0 (carriage return without line feed) is ignored on 7-hole and card output and correctly done on the line printer and 5-hole tape. Compound characters can therefore be printed on the Anelex and the teleprinter by overprinting.

1.4 Selecting Input and Output

The data required by a program may arise from several different sources and each such batch of data may be prepared as a separate unit called a document. Each document will be on a separate paper tape or deck of cards and each will be allotted an input number as described in Chapter 10 of the Atlas Provisional Programming Manual (CS 348). Similarly if several distinct sections of output are to be produced by the program each may be given an output number.

When the program requires input from a given document it first selects the number of that document by extracode. Similarly to send information to a given output stream, the output number must first be selected. The extracodes for these purposes are given below; each one is singly modified.

1050 Select Input n

All succeeding input operations, until the next 1050 instruction, refer to Input n.
If no input document with number n has been defined there will be an exit to the monitor routine.
If input instructions are obeyed without previously selecting an input, Input zero is used (see Chapter 10, Atlas Provisional Programming Manual)

1051 Find Selected Input

ba' = number of currently selected input.
This instruction is particularly useful in subroutines. The current input may be stored at the beginning of the routine by

1051 6 0 0 $b6^1$ = select input.

Another input may then be selected and the original one restored at the end of the subroutine by the instruction

1050 0 6 0 Select input b6

1060 Select Output n

All succeeding output instructions, until the next 1060 order, refer to output n.
For internal code output n is written without an octal fraction (or with an even octal fraction which will be disregarded). If binary output is required n should have an odd octal fraction (usually .1). Thus

1060 0 0 3.1

will select output 3 for binary output.
If output n has not been defined there will be an exit to the monitor routine. If output instructions are obeyed without previously selecting an

output, output zero is used (see Chapter 10, Atlas Provisional Programming Manual)

1061 Find Selected Output

ba' = number of currently selected output plus octal fraction as in 1060
This extracode is used in a similar fashion to 1051

1.5 Input using L100

Since input and output require fairly complicated programs it is usually convenient to use library subroutines for these purposes. L100 is the Input Library Routine. It will reconstruct a record and present the programmer with a number or a character. It will also read texts.

1.5.1 Line Reconstruction L100 itself calls in L199 to reconstruct records from the system input tape. The record or 'line' is reconstructed as follows:

a) Each character is stored at the least significant end oo of a half word.

b) Shifts to inner and outer sets will not be stored. Instead each simple character will be allotted seven bits. The most significant bit will be zero if the character is a member of the inner set and a one if it belongs to the outer set. Thus 'M' is stored as K5.5 and 'm' as K15.5.

c) The character backspace (BS) will be correctly interpreted but not stored. That is, characters before and after a BS will be combined to form a compound character (see 1.6.3)

d) The character tabulate (TB) is not stored. Instead the correct number of spaces will be inserted (see 1.7.6)

e) The last character of a line will be the carriage control character.

f) The following characters have special representations instead of their normal internal code values:-

Space - SP is stored as 0.0
Erase - ER is stored as J4
Fault - FT is stored as J3
Underline - UL is stored as J2
Figure Shift - FS is stored as J1 (two or more successive figure shifts in the same record are stored as a single J1)

Normally only two lines may be reconstructed in this way at one time. Thus only two inputs may be active at the same time (but see 1.7.4)

1.5.2 Entries to L100 Several entries are provided to L100 to enable the programmer to read numbers or characters from the reconstructed line. If numbers are read they will be properly translated into forms acceptable by the accumulator or a B-line.

For numerical input the character erase (ER), or any compound character containing ER, will be ignored. For character and text input ER is read.

For all entries to L100, the link must be stored in B90.

A list of entries is given here; further details are to be found in 1.6.

A1/L100	am' = next number QR
A2/L100	b81' = next integer
A3/L100	b81' = next character
A4/L100	Lose the rest of the line
A5/L100	Read text to store line b89 onwards
A6/L100	Read text, following T newline or T/newline to store line b89 onward
A7/L100	b81' = next integer as a 24 bit integer
A8/L100	b81' = next integer plus 3 bit octal fraction

1.5.3 Data Preparation for L100. The following rules must be obeyed when punching data for L100. Throughout this section

a and c are decimal integers
b is a decimal fraction
and k is a one digit octal fraction.

a) A maximum line length of 160 characters plus a carriage control character is allowed.

b) All numbers must be punched in the form:-

< Layout characters>< Number ><Terminator>

The layout character may consist of any combination of

spaces,
tabs,
new lines,
paper throws,
upper and lower case shifts,
carriage returns,
line feeds,
figure shifts
or back spaces.

No layout character is really necessary.
The terminator may be a

space,
tab,
new line (or end of card)
carriage return followed by line feed,
comma,
paper throw,
or line feed.

c) Numbers to be read by entries A1/L100 will normally take the form

$\pm$ a.b

where the '+' may be omitted if desired. If either a or b is zero it may be omitted and if b is omitted the decimal point is optional. Example:-

10 +7 -4 31. +167 -8.
.17 +.3 -.761 34.61 +0.357 -26.4

Numbers in floating point form will also be accepted by L100 with these two entries. These must be in the form

$\pm$ a.b($\pm$c)

where c is a decimal exponent. a and b are as before; c must be preceded by a '+' or '-'. No spaces may occur between b and the character 'c' or within the brackets.

For example

3.141(+7)

will be accepted. It has value 3.141×10^{7}

d) For entry at A2/L100 or A7/L100, numbers must take the form

$\pm$a

where the '+' is optional and $a \leq 2^{21}-1$ for A2/L100 and $a \leq 2^{24}-1$ for A7/L100
Example:-

64 +731 -2

e) For entry at A8/L100, numbers must take the form

$\pm$a.k

where '+' is optional and $a \leq 2^{21}-1$. If either a or k is zero they may be omitted and if k is omitted the octal point is optional.
Examples:-

14 +2 -51 12. +21. -741.
.7 +.2 -.5 61.1 +7.0 -0.4

f) Texts for input by entry at A5/L100 will normally be punched as one record ending with a carriage control character. If a text takes up more than one record (i.e. more than one line of paper tape or more than one card) the characters ((must be punched at the end of each record except the last.

g) Texts for input by entry at A6/L100 must be preceded by T/newline (a warning character other than T may be used by the programmer).

1.5.4 Punching Errors.

a) If the routine meets an unacceptable character when searching for the beginning of a number, control is switched to A21/L100. For example this would happen on meeting Z in the list of integers

121 316 -7 Z

A21/L100 is normally a fault routine within L100 but can be a private fault routine set by the programmer (see 1.7.2).
The disallowed character will be in B82.

b) If the routine meets an unacceptable character in the middle of a number, control is switched to address A22/L100. Again, the programmer may set A22/L100 for private action and the disallowed character is in B82 as before. Such an entry will be caused for example, by attempts to read a number with a fractional part by entry A2/L100. (See 1.7.3)

1.6 The Entries to L100 in Detail

All entries apply to the currently selected input.

1.6.1 A1/L100.

A number is assembled from the line reconstruction and is stored as a floating point number in Am. It is rounded in the normal Atlas fashion and standardised, and L is cleared.

The terminator is given in B82. If the terminator is also a carriage control character the most significant bit of B82 is set as one. Thus, when the last number of a line has been read, B82 will be negative.

For example if the number is terminated by a space, b82 will equal 0.0. If the terminator is new line however B82 will contain J4 K2.1.

Example

Read ten numbers from input 2 and store them in floating point form in locations 40 to 49.

```
    121     3     0     -9          b3' = count
    1050    0     0     2           select input 2
  5)1101    90    0     A1/L100     Enter L100 to read to Am
    356     0     3     49          Store number
    201     127   3     A5          Count and return.
```

1.6.2 A2/L100. A 21-bit integer is formed from the reconstructed line and stored in bits 0 to 20 of B81; bits 21 to 23 are cleared. The terminator is given in B82 in the same way as with entry A1/L100.

Example

Read to the end of the line of integers on input 1 and store them in half-words from 12.4 onwards.

```
     121    1     0    0             b1' = count
     1050   0     0    1             Select input 1
     121    90    0    3*            Set link
   1)124    1     0    0.4           Increase count by one half-
                                     word
     121    127   0    A2/L100       Read next integer to B81
     113    81    1    12.0          Store integer
     216    127   82   A1            Test for carriage control
                                     character
```

1.6.3 A3/L100. A single character or a compound character is taken from the line reconstruction and stored in B81. A single (seven-bit) character goes into bits 17 to 23. The characters ER, FT, UL and FS will have the values given in 1.5.1.(but SP = 0.1)

A compound character arises from seven-hole tape input whene two or three characters are punched, separated by backspaces. In the case of two characters, during line reconstruction, the numerically smaller character is stored in bits 17 to 23 of the half-word; the other character in bits 10 to 16. In the case of three characters the third will be stored in bits 3 to 9 regardless of numerical value. Such a compound character will be read to B81 by the use of entry A3/L100.

A Compound character may also include the underline character. In this case bit one will be a 1 (i.e. J2 will be added). If a compound character consisting of more than three simple characters, apart from underline, is detected during line reconstruction then a fault is registered and the program suspended. Details are printed on output zero.

For example, if the characters

0 BS / BS UL

are punched on seven-hole tape, the compound character Ø̲ will be printed by the flexowriter. This will be reconstructed as follows:-

010	0 000 000	0 010 000	0 001 111
UL	No 3rd character	0 is inner 2.0	/ is inner 1.7

The instruction

```
1101   90   0   A3/L100
```

to read this character would produce

```
                b81' = J2K401.7
```

After the last character of a line has been read, the next entry to A3/L100 causes B81 to be cleared. The carriage control character will be given in B82.

1.6.4 A4/L100. The routine will set up the conditions necessary for line reconstruction on the next entry to L100. In this way the remainder of the currently reconstructed line is lost.

1.6.5 A5/L100. Read the next record(s) from the current input and store it beginning at the half word whose address is given in B89. The number of characters stored will appear in bits 1 to 23 of the half word whose address was specified in B89. The count includes the carriage control character. The final contents of B89 is the address of the half word which immediately follows the stored text. (For texts of several lines see just below).

1.6.6 A6/L100. This entry is to be made to read a text which appears amidst numerical data at a point not necessarily known in advance to the programmer. The text must be preceded by T Newline or T/Newline according as it is desired later to output the text on a new line or on the current line. (In fact any non-numerical warning character can be used in place of T). The text is then read in by entering A6/L100 via the illegal-character-in-place-of-a-number entry A21/L100 (see 1.7.2) The latter label must therefore be set by the programmer. For example if the following data are being read by A2/L100

```
                1, 2, 3, T
                TEXT
                4, 5, 6
```

then the instructions

```
                21/L100) 121    89    0    3:
                         1362   0     0    A6/L100
```

will store two records when the T is met. The first record stored will consist of a single new line character and the second will be the entire record which follows the T on input. Each record is stored with its character count as described in 1.6.5. The count for the first record has bit 0 equal to 1 as required by the entry A6/L1 to print the text.

Upon exit from A6/L100 B89 is set to the address of the next available half word following the stored text.

On entry to A5/L100 or A6/L100 for a text consisting of one record, the next record from the system input tape will be taken and placed in the store beginning at the address specified in B89. This must be a full word or half word address and the first half word will be used to hold a count of the characters in the record including the carriage control character. The count will be in the character position of the half-word so for a text of 14 characters the count will be 1.6.

If the continuation mark ((has been used each of the records of the text will be read to store locations beginning at b89. Each record will be preceded by a half word containing a count of the characters and each of these half words except the last will also have bit zero set to one. The text will consist of all characters up to but not including the characters ((, and these characters must be followed immediately by new line.

For example, suppose a text on seven-hole tape consisted of the two lines

```
TAPE 12 ((
BLOCK 3
```

The instructions

```
121     89     0     100
1362    0      0     A6/L100
```

would set the contents of half words 100 onwards as follows:-

```
(100)      = J4K1.1        9 characters including 2 SP and one
                           NL plus a continuation mark
(100.4)    = K6441604.5    TAPE
(101)      = K0121220.1    Sp 12 Sp
(101.4)    = J21           NL
(102)      = K1.0          8 characters including one SP and
                           one NL.
(102.4)    = K4254574.3    BLOC
(103)      = K5301232.1    K sp 3 NL
```

1.6.7 A7/L100. An integer is formed from the line reconstruction and stored in B81 with its least significant digit as bit 23. The terminator is given in B82.

1.6.8 A8/L100. A 21-bit integer with one octal digit after the point is formed from the line reconstruction and stored in B81. The least significant digit of the integer occupies bit 20 and the octal digit occupies bits 21 - 23.

The terminator is given B82.

1.7 Optional Parameters of L100

Library routine L100 contains seven parameters which are optionally set. These may be given alternative values by the programmer if he wishes. If L100 is called for explicitly by an L directive these parameters must be set first. This for example allows ABL to leave the right amount of space in the complied program to contain L100. If L100 is called for implicity the parameters may be set at any point in the program. In this case L100 is stored somewhere after the compiled program and the exact space it will occupy is irrelevant at the time of the implicit setting.

The parameters are:-

A20/L100 - beginning of line reconstruction storage
A21/L100 - Routine for fault at beginning of number
A22/L100 - Routine for fault in the middle of a number
A23/L100 - Maximum number of active input streams
A24/L100 - Maximum line length in characters
A25/L100 - Tab settings
A26/L100 - Tab routine

These are dealt with individually below.

1.7.1 A20/L100. The programmer may allocate specific storage for the beginning of line reconstruction by setting A20/L100, e.g. A20/L100 = 1000. If A20/L100 is not set by the programmer A20 will follow L199.

1.7.2 A21/L100 If during numerical input a spurious character is encountered instead of a number, control is switched to A21/L100 with the character in B82. The programmer can write his own fault routine to deal with such a situation. This is particularly useful in dealing with a number list of unknown length.

Suppose the list is punched out and the terminator of the last number is followed by the character '*' (internal code 1.6). The following piece of program would read the numbers from input 2 and form their sum in A5. If a spurious character other than * is met (owing to mispunching) control is transferred to store line A6.

```
        1050      0       0      2           Select input 2
         346      0       0      J4          Clear am
      4)1101     90       0      A1/L100     am' = next number
         320      0       0      A5          am' = partial sum
         356      0       0      A5          (A5)' = partial sum
         121    127       0      A4          Go to read next number
        5)+0
21/L100)172      82       0      1.6         bt' = b82 - '*'
         225    127       0      A6          Go to fault routine)
        - - - - Next Instruction             if not *           )
```

If A21/L100 is not set by the programmer a standard fault routine will be entered.

1.7.3 A22/L100. If during numerical input a spurious character is encountered within a number, control is transferred to A22/L100 with the character in B82. The programmer may write his own fault routine to take care of this situation but he will have a partially assembled number to deal with.

If A22/L100 is not set by the programmer a standard fault routine will be entered.

1.7.4 A23/L100. The maximum number of inputs active at one time (i.e. those with a line reconstructed part of which remains to be read) is normally two. The programmer can, if he wishes, alter this by setting A23/L100 to the number he requires. Thus the directive

A23/L100 = 4

would permit four streams to be active at once. This would obviously involve more store being used by the line reconstruction routine to accommodate the reconstructed lines.

1.7.5 A24/L100. The maximum line length accepted by L100 from any peripheral is 160 characters (excluding carriage control information). If the programmer wishes to use a different line length he must set A24/L100 to the number of characters he requires.

1.7.6 A25/L100. This parameter is optionally set within the library routine to 15 which gives the standard tab settings of

8, 8, 8, 8, 16, 16, 16, 16 . . . .

By setting A25/L100 to -1 or 999 the programmer can arrange for tabs of 16, 16, 16, ... or 8, 8, 8, ... respectively.

1.7.7 A26/L100. During line reconstruction a standard routine for replacing 'tab' by the correct number of spaces begins at location A26/L100. If the programmer wishes to use a private routine to deal with tab, A26/L100 must be set to the starting address of this routine. Exit from the private routine must be to 1A28/L199

1.8 Output using L1

L1 is the Output Library Routine and will output a number from the accumulator, an integer from a B-line, a single simple or compound character, or a group of characters forming a text. In actual fact this routine transfers the output information to the System Output Tape whence it is automatically sent to the required peripheral when output is completed. For this reason the single output routine can be used regardless of the output peripheral.

1.8.1 Entry points to L1. Different entry points are provided for each type of output required and the way in which numbers are to be output is specified by a style number in a B-line before entry to L1.

For all entries the link is stored in B90.

The entry points are as follows:-

A1/L1 Outputam in style b89
A2/L1 Output b81 in style b88
A3/L1 Output one character from b81
A4/L1 End line (or card)
A5/L1 End record by carriage control character in B87
A6/L1 Output a text from b89 onwards
A7/L1 Output a text from b89 onwards, with terminating
carriage control character from B87

Further details of each entry are given below.

1.9 The Entries to L1 in Detail

All entries apply to the currently selected outputand length is limited only by the output peripheral.

1.9.1 A1/L1 Entry to L1 at this point causes L to be cleared and the contents of the accumulator to be output as a fixed or fl floating point number. The style of output is determined by the contents of B89, which must be set before entry to L1 by an order of the form

```
121    89    0    p:q.k
```

Here p is the number of decimal digits required before the point $(0 \leq p \leq 127)$
q is the number of decimal digits required after the point $(0 \leq n \leq 15)$
and k (the octal digit) indicates the form in which the number is to be printed as follows:-

k = 0 Accumulator printed fixed point, signed, on same line
1 Accumulator printed floating point, signed, on same line
2 Accumulator printed fixed point, signed, on new line
3 Accumulator printed floating point, signed, on new line
4 Accumulator printed fixed point, unsigned, on same line
5 Accumulator printed floating point, unsigned on same line
6 Accumulator printed fixed point, unsigned, on new line
7 Accumulator printed floating point, unsigned, on new line

If k is zero, .k may be omitted.

Further details are as follows:-

a) All numbers will be correctly rounded to the last digit printed. The rounding is decimal and of the 'add 5' variety.

b) The contents of the accumulator will be spoiled, but the contents of B80, B88, B89 are preserved. "On the other hand B81 through B86 are destroyed".

c) When k takes the value 2, 3, 6 or 7, the current line is terminated by a single new line character before the output of the number. Otherwise the number follows the last character of the current line.

d) For k = 0 to 3 the number is printed signed, that is preceded by Sp for positive numbers or - for negative numbers (but see 1.10.2). With k = 4 to 7 the Sp or - are omitted altogether; for example both +2.5 and -2.5 would be printed 2.5.

e) If k is odd the number is output in floating point decimal form. The mantissa is printed with one non-zero decimal digit before the point and q digits after (i.e. $1 \leq$ mantissa < 10) and is followed by the exponent. Zero in floating point form will have mantissa of 0. followed by q zeros and an exponent of +0. The number of character spaces to precede the decimal point is p+1 for signed numbers or p for unsigned numbers.

The floating point number appears in the form

mantissa (exponent)

with a two decimal digit exponent preceded by + or - and enclosed in brackets (but see 1.10.4 and 1.10.5).

A non-significant left hand zero of the exponent will be omitted and a space will be output after the final character (i.e. the ')'). If the exponent is more than two digits it will be printed in full but layout will be spoiled (but see 1.10.4).

f) If k = 0, 2 the number is printed in fixed point form with p+1 character positions before the point and q after; if k = 4, 6 there are p character positions before the point and q after. Left hand zeros of the integral part are replaced by spaces; right hand zeros of the fractional part are always printed. Positive numbers are printed without a sign.

If the number has more than p places before the decimal point (say P places) it will be printed either as a fixed point number with P places before and q after, or as a floating point number with (p+q) significant figures in the mantissa, whichever form has fewer characters. On either case layout will be spoiled.

g) The input routine L100 uses index registers B88 and B89. Care must therefore be taken not to spoil the style set for an L1 entry by first entering L100.

h) The special cases of p = 0 and q = 0 are dealt with as follows:-

p = 0
The integral part of the number in fixed point and of the mantissa in floating point will appear as 0. Thus 0.25 output in style 0:2 would appear as

sp 0.25

while -1.25 output in style 0:3.1 would be printed as

-0.125 (+1) sp

q = 0
An integer with no decimal point will be printed.

p = q = 0
The number is printed as sp0 or -0 with a decimal exponent if necessary.

Examples

1. Print the numbers stored in locations 60 to 79 on output 3. The numbers should be printed signed in floating point form, each on a new line, with four decimal places after the point.

```
   1060    0      0     3        Select output 3
   121     1      0     -19      b1' = modifier
   121     89     0     1:4.3    Style in B89
 1)324     0      1     79       Read next no. to Am
   1362    0      0     A1/L1    Print a
   201     127    1     A1       Count and return
   - - - - Next Instruction
```

2.   Read the next number from input 2 and print it on output 1 on the current line.   The number should be fixed point, unsigned with four places before and two places after the decimal point.

```
1050    0     0    2          Select input 2
121     90    0    2*         Set link
121     127   0    A1/L100    Read one number
1060    0     0    1          Select output 1
121     90    0    3*         Set link
121     89    0    4:2.4      Set style
121     127   0    A1/L1      Print number
- - - - Next Instruction
```

1.9.2 A2/L1.   Entry to A2/L1 causes a 21 or 24 bit integer from B81 to be printed.   The style of output is determined by the contents of B88 which must be set before entry to L1 by an order of the form

```
121    88    0    p:q.k    .
```

Signed integers are printed with p+1 character positions before the point, if any (see b) below), and unsigned integers with p character positions before the point, if any.   Left hand zeros of the integer are replaced by spaces.   Positive signed integers are preceded by a space rather than a plus sign (to print something other than a space, see 1.10.2).

k is interpreted as for accumulator output except that the distinction between fixed and floating point numbers is replaced by that between 21-and 24-bit integers.   Thus the interpretation of k is as follows:-

k = 0   b81 is printed as a 21-bit integer, signed, on same line
    1   b81 is printed as 24-bit integer, signed, on same line
    2   b81 is printed as a 21-bit integer, signed, on new line
    3   b81 is printed as a 24-bit integer, signed, on new line

k = 4   b81 is printed as a 21-bit integer, unsigned, on same line
    5   b81 is printed as a 24-bit integer, unsigned, on same line
    6   b81 is printed as a 21-bit integer, unsigned, on new line
    7   b81 is printed as a 24-bit integer, unsigned, on new line

Further details are:-

a) The contents of the accumulator and of B81 are spoiled, but the contents of B87, B88 and B89 are preserved. B80 is not used but B82-86 are overwritten.

b) If k is even b81 is treated as a 21-bit integer.   If q = 0, bits 0 to 20 are taken as a pure integer and bits 21 to 23 are ignored.   If q takes any non-zero value the integer is followed by a point and one octal digit taken from bits 21 to 23.

c) If k is odd b81 is printed as a 24 bit integer. The value of q is irrelevant.

d) If the integer has more than p digits it will be printed correctly but layout will be spoiled.

Example

Print the number 97 on a new line on output 1 followed by the contents of store half-word 97 as a 21-bit integer with octal fraction. Allow for six characters before the octal point of the integer.

```
1060    0     0     1        Call
121     88    0     2:0.6    Style in B88
121     81    0     97       b81' = 97
1362    0     0     A2/L1    Print '97'
121     88    0     5:1.0    Style in B88
101     81    0     97       b81' = (97)
1362    0     0     A2/L1    Print (97)
```

1.9.3. A3/L1. Entry at this point will cause one character, which may be simple or compound, to be printed from B81. Simple characters must be placed in the least significant six bits of B81 with bit 17 indicating whether the character is inner or outer set as 1.6.3.

An alternative way of printing an outer set character is by outputting the 'shift to Outer Set' character, K0.4. All succeeding characters are then specified by six bits only and will be printed as outer set characters until the 'shift to Inner Set' character, K0.5 is encountered or the end of the record is reached. Thus to print outer set characters from the next record a further 'shift to Outer Set' character must be output.

Compound characters may also be built up in B81 as described in 1.6.3. Entry at A3/L1 will unpack each character and output them separated by backspaces. (This facility can be employed only if output is to 7-hole paper tape since this is the only output medium with a BS character). Underline characters are allowed in the same way as shown in 1.6.3.

The characters ER, FT, UL and FS may take either the forms adopted by L100 (i.e. J4, J3, J2 and J1) or their internal code numbers (K17.7, K7.7, K12.6 and K0.7). The fault character, which normally arises only from input via L100, is output as erase underlined. The actual printing of the fault character on the different peripherals is as follows (see 1.2)

7-track	ER
5-track	ER ER ER
Analex	.. ‾
Cards	.. _

All compound characters containing J4 (i.e. with bit zero set to one) will be output as erases. If an attempt is made to print an impossible character (e.g. [on 5-hole) a full stop will be printed instead.

Example

Print, on the current line of output 6, which is a seven hole paper tape punch, the characters

```
                Ab Ø

          121     81     0     K4.1          b81' = 'A'
          1362    0      0     A3/L1         Punch A
          121     81     0     K14.2         b81' = 'b'
1         1362    0      0     A3/L1         Punch b
          121     81     0     K0.1          b81' = 'SP'
          1362    0      0     A3/L1         Punch SP
          121     81     0     J2K401.7      b81' = Ø
                                             Punch / BS 0 BS UL
```

While entry A3/1 is useful for the output of compound characters, it is more efficient to use the extracode 1064 to output single characters (see 1.14).

1.9.4 A4/L1. With this entry, the current line of output is terminated by a single new line (or end-of-card) character. The extracode 1065 0 0 2.1 has exactly the same effect and should normally be used.

1.9.5 A5/L1. If more than one new line character, or some other carriage control output is required, entry may be made to A5/L1. B87 must contain the carriage control information as detailed in 1.3.

Thus for six new lines the necessary instructions are

```
121     87     0     K2.6
1362    0      0     A5/L1
```

The extracode 1065 ·0 87 0 has the same effect, and should normally be used.

1.9.6 A6/L1. Entry to A6/L1 will output a text from store locations beginning at b89. The text must be stored in the form in which texts are read by L100, (see 1.6.6). That is

a) each record of the text must be preceded by a half word containing a count of the number of characters in the record including the carriage control character.

b) for each record except the last this half word must also have bit zero set to one (i.e. for n characters, the half word must contain YnJ4).

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

The text stored by L100 in the example at the end of paragraph 1.6.6 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.
```

1.9.7 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.

1.10 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 exponent.
A29/L1	

Details are given below.

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

1.10.2 A25/L1. As stated in 1.9.1 and 1.9.2 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.

1.10.3 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.

1.10.4 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.

1.10.5 <u>A28/L1 and A29/L1.</u> These characters 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 3.16×10^{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.

1.11 X <u>Input and Output by Extracode</u>

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.

1.12 Binary Input and Output

Input and output extracodes may work with either the 12-bit binary characters described in 1.1.4 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 not 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 marker ***B, etc., then even parity characters will be accepted and stored as for odd parity characters.

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

$$\underline{\pm}\,\underline{-}\,\underline{0}\quad \underline{1}\,\underline{2}\,\underline{3}\quad \underline{4}\,\underline{5}\,\underline{6}\quad \underline{7}\,\underline{8}\,\underline{9}\quad .$$

For output, as described in section 1.4 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.

1.13 The Input Extracodes

The Select Extracodes have been described in section 1.4 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 extraode 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 records 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 extracode 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.

Example

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 charactersfour 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 2) if next.character
                                        is binary
        1054    60      0     -1*      )Read next ch. to B60
                                       )Check next record if carriage
                                        cont. ch.
        113     60      1     -0.4A5    Store character
        200     127     1     A1        Go to 1) 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
        113     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
```

1.14 The Output Extracodes

The necessary extracodes for selecting output streams have been described in section 1.4. 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 Documents)

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 reglected 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 when 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 locatipns
                                        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 DOCUMENTS. 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.
For example, to define a card punch output with number six to which a maximum 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
```

1.15 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 K2.1, i.e. 010.001

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 recognised 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.