SECTION 7 - INPUT AND OUTPUT

7.1. Input and Output Instructions - Buffers

This section contains details of the input and output instructions and the operation of the buffers that are common to all peripheral units. Aspects that are peculiar to the individual units are dealt with in the appropriate sections.

7.1.1. Functions

The function number in input and output instructions gives the type of conversion that is required, and is independent of the type of unit being addressed.

The functions are:-

17 - Block input
18 - Block output
19 - Input with decimal conversion
20 - Input with sterling conversion
21 - Alphanumeric input
22 - Output with decimal conversion
23 - Output with sterling conversion
24 - Alphanumeric output
28 - Output with spaced sterling conversion

7.1.2. Characters

Transfers of information between the computer and the peripheral units (both input and output) are in terms of 'characters'. Each character consists of 6 binary bits, and it may have an individual meaning according to a laid-down code, or it may simply be part of a binary computer word which is to be recorded on magnetic tape and later read back and re-assembled.

Details of the 6-bit character code are given in Appendix I.

7.1.3. Channels

Input and output instructions address the peripheral unit they require by a 'channel number' between 1 and 15. There is a conventional allocation of channel numbers to various types of peripheral unit, but the connections are in fact made through a plugboard which allows the arrangement to be easily changed if desired.

The conventional allocation is as follows:

1 - 6	Magnetic tape units
7, 8	Card readers
9, 10	Printers
11, 12	Paper tape readers
13	Card punch
14, 15	Paper tape punches.

The earlier numbers of each set are used first, so that for example four tape units are given channel numbers 1 - 4, and one card reader is given channel 7.

7.1.4. Layout of Instructions

Input and output instructions all have a similar layout. Besides the function, the parts of an instruction are used as follows:-

A address - source or destination register (first register in case of block transfer).

C1 address - number of characters to be transferred (zero for block transfer).

C2 address - channel number.

The B address is not used in simple input or output transfers. Its purpose is to give control signals to the peripheral units and buffers; details of these are given in the descriptions of the units, and below in the paragraphs on buffers (15 and 16).

7.1.5. Input Instructions

7.1.5.1 Function 17 - Block

Read 96 characters from the channel given in C2, assemble them into a 16 word block and place in the 16 consecutive registers starting at register A. The first character occupies the most significant position in the first word.

7.1.5.2. Function 19 - Decimal

Read the number of characters (up to 15) given in C1 from the channel given in C2, convert from decimal to a binary number and place in register A. (See 8.1; 9.1; 11 of this section).

7.1.5.3. Function 20 - Sterling

Read the number of characters (up to 15) given in C1 from the channel given in C2, convert from sterling to binary pence and place in register A. The last three characters are taken to be tens of shillings, shillings, and pence. (See 8.1; 9.1; 10; 11 of this section).

7.1.5.4. Function 21 - Alphanumeric

Read the number of characters (up to 12) given in C1 from the channel given in C2, make up to 12 characters with 'no character' symbols if C1 < 12, and place in registers A and A + 1. The last character read in occupies the least significant position in A + 1. (See 8.2).

7.1.6. Output Instructions

7.1.6.1. Function 18 - Block

Take the contents of the 16 consecutive registers starting at register A, break them down into 96 characters and put out to the channel given in C2. The most significant character of register A is sent out first.

7.1.6.2. Function 22 - Decimal

Take the contents of register A and convert them to a decimal number. Put out the number of characters (up to 12) given in C1 to the channel given in C2; the characters are counted from the less significant end of the converted number, but the most significant of them is put out first. (See 7; 8.2; 9.2).

7.1.6.3. Function 23 - Sterling

Take the contents of register A and, regarding them as binary pence, convert them to a sterling number. Put out the number of characters (up to 12) given in C1 to the channel given in C2; the characters are counted from the less significant end of the converted number, but the most significant of them is put out first. (See 7; 8.2; 9.2; 10).

7.1.6.4. Function 24 - Alphanumeric

Take the contents of registers A and A + 1 and split them into 12 characters. Put out the number of characters (up to 12) given in C1 to the channel given in C2, counting the characters from the less significant end of A + 1 but putting out the most significant first. (See 8.2).

7.1.6.5. Function 28 - Spaced Sterling

Take the contents of register A and, regarding them as a binary number of pence, convert to a sterling number. Insert 'space' characters between pounds and shillings and between shillings and pence. Put out the number of characters (up to 12) given in C1 to the channel given in C2; the number of characters includes the spaces and is counted from the less significant end of the converted number, but the most significant of them is put out first. (See 7; 8.2; 9.2; 10).

7.1.7. Zero Suppression

In decimal and sterling output instructions the process consists of a complete conversion of the binary number followed by a procedure to put out the required number of characters. Non-significant zeros (i.e. those to the left of the first non-zero character of the converted number) are changed to 'space' characters in the output procedure. This change is made without reference to the number of characters specified in the C1 address. In sterling output instructions a zero in the tens of shillings position is always suppressed. A zero number is output as a single zero in the least significant position, preceded by spaces up to the number of characters called for.

Examples (* indicates a 'space' character) :

Function	C1	Converted Number	Output
22	10	000002027539	***2027539
22	6	000002027539	027539
23	8	000000527139	**527139
23	8	000000527039	**527*39
28	8	000000527039	527**3*9
22	6	000000000000	*****0

7.1.8. Number of Characters

7.1.8.1. Functions 19 and 20

Any number of characters from 1 to 15 may be called for from the input. (If 0 is specified, one character will be read). If the resulting binary number exceeds one word

(36 bits) then the most significant bits will overflow and be lost.

7.1.8.2. Functions 21, 22, 23, 24 and 28

Any number of characters from 1 to 12 may be specified. If 0 is specified twelve characters will be transferred, and if 13, 14 or 15 are specified none will be transferred.

7.1.9. Negative Numbers

7.1.9.1. Input

A minus sign (6-bit character value 34) is detected by the input conversion functions (19 and 20) and causes the resulting binary number to be complemented before being stored. The minus should precede the number to which it applies, and must be included in the number of characters called for.

7.1.9.2. Output

The output conversion functions (22, 23 and 28) do not print negative binary numbers as sign and modulus, but treat them as large positive values. If a negative result is possible, the program should test for the occurrence, output a minus sign as a separate process, and complement the number before using the output conversion instruction.

7.1.10. Binary Halfpence

The sterling functions (20 and 23) can easily be altered to give conversion to and from binary halfpence instead of binary pence. In this case the least significant character on input is the halfpenny, and may be given as 1 or ½ if present and 0 or 'space' if absent. On output the least significant character will be ½ or 'space'.

Function 28 may not be used with halfpence conversion.

7.1.11. Note on Input Conversions (Functions 19 and 20)

If 6-bit characters other than digits 0 - 9, 'space' or 'no character' (which are treated as zero), and 'minus' occur in the input during conversion, they are treated as follows:-

(i) characters with 6-bit values 10-31 are added into the converted number with appropriate significance, e.g.

275&06 input by function 19 will give the binary equivalent of 276506 (& = 15).

(ii) certain 6-bit codes act as minus signs, causing the converted number to be complemented. These are 38, 42, 46, 50, 54, 58 & 62 (A, E, I, M, Q, U & Y).

(iii) 6-bit codes in the range 32-63 cause certain values to be added into the converted number with appropriate significance, according to the following table.

Code	Meaning	Value	Code	Meaning	Value
32	'No character'	0	48	K	16
33	½	1	49	L	17
34	Minus	0	50	M	16
35	Space	00	51	N	16
36	Buffer Control	4	52	O	20
37		5	53	P	21
38	A	4	54	Q	20
39	B	4	55	R	20
40	C	8	56	S	24
41	D	9	57	T	25
42	E	8	58	U	24
43	F	8	59	V	24
44	G	12	60	W	28
45	H	13	61	X	29
46	I	12	62	Y	28
47	J	12	63	Z	28

Examples:

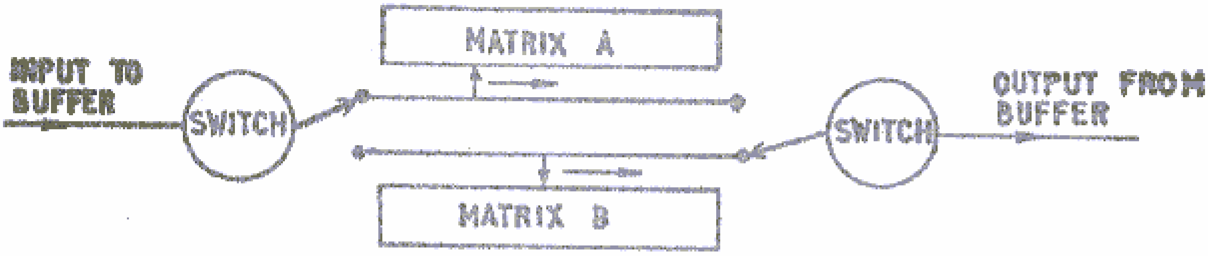
ABCD input by function 19 gives the binary equivalent of - 4489.

27P3 input by function 20 (binary pence) gives the equivalent of 2 pounds 91 shillings and 3 pence, i.e. 1575 pence.

7.1.12 Buffers

Each peripheral unit transfers information into or out of the computer through a buffer store. The purpose of these buffers is to match the speed of the units to that of the computer and to even out irregularities in timing. The size and arrangement of buffer differs from one unit to another, and details will be found under the appropriate sections.

In all units except the 25 character per second paper tape punch, the buffer contains two 'matrices'. These are used alternately, as shown in the following diagram:



All information entering or leaving the buffer is in the form of 6-bit groups or 'characters', (and it is held in the buffer in this form). As shown, characters entering the buffer at the left-hand side are placed in consecutive positions in matrix A, indicated by a pointer which moves from left to right. Matrix B also has a pointer moving from left to right indicating characters to be sent to the output. Characters therefore leave a buffer in the order in which they entered it.

When a pointer reaches the right-hand end of its matrix, it is returned to the left-hand end again and a signal is given to show that this has happened.

When signals have been given from both matrices, the switches simultaneously change over. Matrix B is now free to receive information from the input, and that previously stored in matrix A can be passed to the output.

7.1.13. Input and Output Buffers

With an input unit, for example a card reader, the input to the buffer is from the reader and the output goes to the computer. With an output unit, for example printer, the input to the buffer comes from the computer and the output goes to the printer.

An input unit is arranged to try to keep its buffer full; an output unit tries to keep its buffer empty. In this way the computer has data in its buffers on which to calculate, and output buffers free to receive results, provided that the speed of the units concerned is not exceeded.

A magnetic tape unit may be required either to read or to write. Its buffer will be connected accordingly, and while it is reading will act as an input unit, while writing as an output unit.

7.1.14. Requests

The peripheral units are arranged to attempt to keep their buffers full (input) or empty (output), and provided they do so all is well. However, the computer program may be able to read data faster than the input unit can provide it, or produce results faster than the output unit can dispose of them. In such a case the computer is limited by input or output, and there is a mechanism in the buffer to ensure that it is kept waiting as long as (though no more than) necessary.

Each input or output instruction causes a 'request' to be sent to the buffer of the unit concerned. If the buffer is in a state for the transfer to take place, a 'reply' is sent back and the input or output operation proceeds. However, if the buffer is not ready, the reply is delayed until the peripheral unit has caught up sufficiently.

7.1.15. Buffer Switching Signal

It may be unnecessary for the computer to completely fill or empty a buffer matrix. To meet such cases, a means is provided of moving the pointer which indicates position in the matrix, to the end, without further characters being transferred. (See para. 12). A '1' in the B address of an input or output instruction causes the pointer to be moved on after the transfer of information has been carried out.

Since the moving on of the pointer gives rise to the signal to switch from one matrix to the other, this digit (which is known as 'D18' owing to its position in the instruction) is also referred to as the 'buffer switching signal'.

7.1.16 Continue Operations Button

As already stated in 7.1.14: if a buffer is not ready to transfer data, the reply signal is delayed. This is known as a machine 'hold up', and will be cleared automatically when the peripheral unit has caught up.

However, if a peripheral unit has been addressed in error by incorrect channel selection, or a program fault, it may not be possible to clear the resulting hold up by adjusting the unit concerned. For this reason, a 'continue operations' button has been provided on the main bay of the computer. The use of the continue operations button causes an artificial reply signal to be generated, so that the required data transfer can be carried out with the following results:

(i) Output instructions will be obeyed, and information will be transferred from the core store and lost. The contents of the core store will not be corrupted.

(ii) Input instructions will be obeyed, and the appropriate locations (1, 2 or 16 words) in the core store will be cleared, irrespective of the input function used.

The hold up must be cleared by adopting the following procedure:

(i) Halt the computer.

(ii) Manually clear the buffer and control of the peripheral unit concerned.

(iii) Push the CONTINUE OPERATIONS button.

It is most important to ensure that the operations are carried out in the correct order, since failure to do so is likely to lead to corruption of subsequent data transfers to and from the computing centre.

7.1.17 Layout Characters

If the B address of an input or output instruction is non-zero, a set of three characters is sent out from the computer when the information transfer is complete. If the buffer is an input one, the characters will have no effect; if it is an output, they will be stored in the next three places in the matrix.

The three characters are first a marker, having the value 36, and then two containing the contents of the B address of the instruction;

D27-24 go in the first and D23-18 in the second.

These layout characters are used by output units when the contents of the buffer are read out. Details will be found under the sections on output units.

7.2. Magnetic Tapes

7.2.1. Specification and Method of Recording Information

Unit used	Ampex FR300	
Running speed of tape	120 in. per second	these figures may vary slightly
Net character transfer rate	7.3 to 7.5Kc/s	

Information is recorded in blocks of 198 characters, each of 6-bits; of these 192 are information characters and 6 are for control purposes. The positions of the blocks are pre-determined, the block markers being recorded before a tape is issued for use. The characters are recorded serially, across the tape, thus:-

```
Block     --> •  1 0 1
Marker           0 1 0
                 1 1 1    etc
                 1 0 0
                 0 1 1
                 1 0 0
      characters 1 2 3  etc.
```

The six information tracks are duplicated. There are, in fact, 16 tracks across the tape, allocated as follows:-

6 information tracks } duplicated
1 clock track }

1 parity track

1 block marker track

The clock tracks are used for timing purposes. The parity track is explained later.

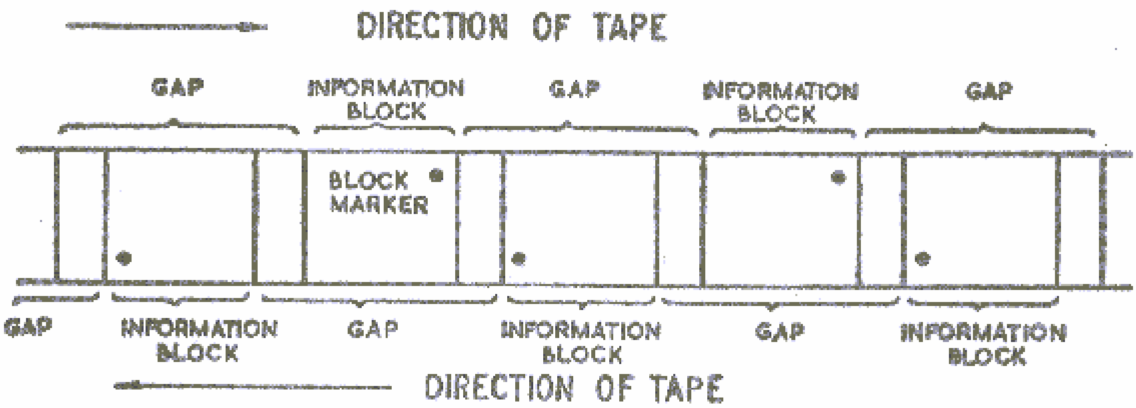
On reading a tape, the 12 information tracks are read simultaneously, and the signals from each pair of tracks are mixed to minimise the chance of a bit being corrupted; if one track fails, the signal from the other is sufficient to give correct information. No provision is included for back spacing or re-reading.

Blocks are recorded in both directions on the tape. The tape runs forward for its whole length, recording or reading on alternate blocks. When it reaches the end, it is automatically reversed, and recording or reading now takes place on the blocks in between. This is done:

(a) to allow space for stop/starting at any point on the tape

(b) to minimise rewinding time (a tape must always be re-wound to its original position after use).

This may be shown as follows:-



Each information block occupies on average 1.2 inches, and the distance between the starting points of consecutive blocks read in the same direction is 3.25 inches (i.e. the pattern is repeated each 3.25 inches). The gaps shown between information blocks are the positions on which the braking rollers act.

Tapes are normally available in five standard sizes. These are shown below, together with the approximate running times and guaranteed minimum capacities:-

Length	300'	600'	1200'	2400'	3600'
Capacity:					
16 word blocks	3,000	7,250	15,750	32,750	50,000
32 word blocks	1,500	3,625	7,875	16,375	25,000
Running time:					
(go and return)	1 min.	2 min	4 min	8 min	12 min

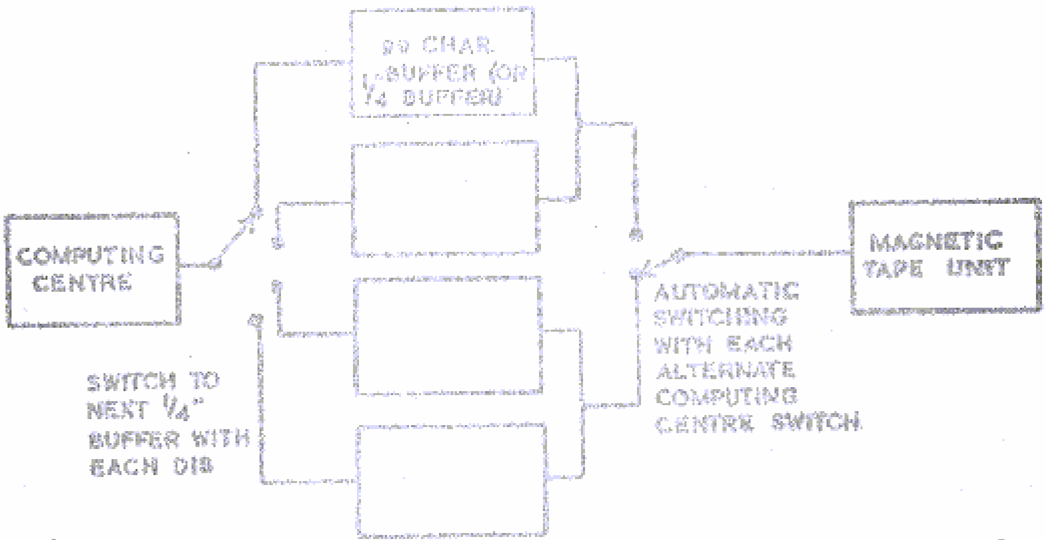
Certain other sizes can be made available by special request.

To maintain full speed on a tape unit, the computer must process the information for, or from, a tape block of 192 characters within 27msecs. If the computer exceeds this, a signal will be given to stop the tape. The unit may, however, be restarted during the slowing down time, so that times slightly in excess of 27msecs. will only cause slight delays.

It is not possible to read from and write (or record) onto a tape during the same run. When a tape is loaded and the unit is started, the tape is automatically only available to be read from. If writing (or recording) is to take place, a special control signal must be sent by program which will set the unit into the 'write' condition. If, subsequently, the tape is to be read back before being removed, another signal returning the unit to the 'read' condition is needed. Both these control signals automatically rewind the tape to the beginning (by the shortest route, i.e. if the tape is moving in the forward direction, it is simply reversed without first running through to the end). A third control signal is used when the unit has been finished with, to rewind and set to 'finish' condition. Rewind, set to 'read' or 'write' operate the tape at normal running speed. Rewind set to 'finish' moves the tape half as fast again.

7.2.2. Buffer system

The magnetic tape buffer system differs from others in that there are effectively four buffers, known as 'quarter buffers', each of 99 characters (96 for information, three for control). Each D18 from the computer switches the computer to the next quarter buffer; the tape unit, however, only moves with each alternate D18, and a tape block of 198 characters comes into, or goes out of, the buffers in one transfer. This may be shown as in the diagram below.



In practice there are two matrices, or '$\frac{1}{2}$ buffers', each of 198 characters; from the programming point of view, however, these may be regarded as shown on page 3. $\frac{1}{4}$ buffers may also be switched, when reading from tape, by omitting the D18 and reading all 99 characters, i.e. the control characters as well as the information characters. This is of use when checking parity, q.v. The transfer of a block of 32 words of information from the buffer to the tape unit when recording, or from the tape unit to the buffer when reading, is entirely automatic.

7.2.3. Program Instructions

The normal functions used with magnetic tape are the block transfers, 17 and 18, for input to, and output from, the computing centre respectively. These functions transfer 96 characters, or 16 words, at a time, without any form of conversion. The information on the tape is thus exactly as it would also appear in the core store, except that it has been broken up into characters, each of 6 bits, and these characters have been recorded serially along the tape. A D18 must be given with each transfer (unless, when reading, parity is being checked).

Example 1.

18 . 101 . 1 . . 3

This instruction transfers the contents of registers 101 to 116 inclusive into the buffer connected to channel 3 (specified in the C2 address), and gives a D18 to switch to the next quarter buffer. The most significant 6 bits of register 101 will appear either as the first character, written across the tape, of a block of 192 characters, or as the 100th character, depending on whether the 96 characters concerned will form the first or second half of a tape block. (It will be noted that three control characters follow each group of 96 characters, so that this first character would not appear in the 97th position). Equally, the least significant 6 bits of register 116 will appear as either the 96th or the 195th character.

Example 2.

17 . A0 . 1 . . 1

This instruction transfers 96 characters from the buffer connected to channel 2 into A0 to A15, and gives a D18 to switch to the next quarter buffer.

7.2.4. Control Signals

To rewind a tape to the beginning, and set it to 'finish', 'read', or 'write' a bit is placed in the D25, 26 or 27 position of the instruction concerned, together with a D18.

Thus:-

1. Rewind the tape on M.T.U. 1 and set it to finish

   19 . 0 . 129 . 1 . 1

2. Rewind M.T.U. 2, set to read

   19. 0 . 257 . 1 . 2

3. Rewind M.T.U. 2, set to write

   19 . 0 . 513 . 1 . 2

It will be seen that the function used is input decimal to register 0 in all cases. In practice any input/output function can be used, but, unless care is taken, other functions can overwrite wanted data, either in the store or on the tape. This cannot happen using an instruction as suggested above, and therefore, by convention, function 19 is usually used. It must be added that, for control purposes only, it is in order to address a deck which is in the output ('write') state with an input function.

Certain rules associated with control signals must be obeyed. They are:-

1. After a rewind, set to write, the tape must be written on (i.e. at least 2 D18's must be given to the deck, so that at least one tape block is recorded) before a further control signal is given to the same deck.

2. After a rewind, set to finish, the tape unit cannot again be addressed by the computer until it has been manually restarted.

3. After a rewind set to read or write, there must be at least one instruction (of any sort) in the program before the next instruction addressing the deck just rewound. This is because the electronics concerned with a rewind takes a little time to absorb the rewind instruction; if the very next instruction tried to address the same deck, the buffer etc. would not be quite ready and the instruction might not be carried out correctly.

4. Before giving a rewind signal to a deck on which writing has been taking place, it is necessary to record one extra (dummy) 16 word block. This is because, when the rewind signal is given, it is carried out as soon as recording from the ½ buffer connected to the tape unit at the time is completed. Should the rewind signal be associated with the second of a pair of ½ buffers, the information in the previous half buffer will be successfully recorded, but the information in the first half of the buffer connected to the computer at the time the control signal is given will be lost. It would be possible to avoid losing information in this way by ensuring that any rewind signal coincides with an odd numbered D13. This, however, means counting blocks, and it is easier always to output the one extra block; if a block is then lost, it will be the dummy block and not the last data block.

7.2.5. <u>Checks on Information</u>

When a tape is read back, it is advisable to have a method of checking that the data have been read correctly. The most usual system is to include with the data, 'check totals' at specific intervals. For example, each 16 words might consist of 15 words of data, and a total of these 15 words in the 16th word. To create such a check total and output 1 x 16 word block to tape, the instructions might be

30 . B0 . B15 . 15

18 . B0 . 1 . . 1

This sums B0 to B14 and puts the complemented total (termed in this case, the 'negative check total') in B15, then outputs the block. When the block is read back, the check total can be checked by writing

```
17. B0 .  1 .    . 1
30. B0 . B16 . 16 .
12. B16.  R-         ——> error routine
```

It will be seen that the positive sum of B0 - 14 added to the negative check total should produce zero.

A second method of checking information read back, less often used, is the parity system. As each character is written onto tape, the machine automatically generates, and records, a 'parity' bit. This bit, which will be a 'one' or a 'zero' as appropriate, makes the total number of 'ones' written across the tape up to an odd number. When the tape is read back, parity is automatically checked, so that, if the machine finds an even number of 'ones' (including the parity bit) parity has failed. If this happens, an asterisk (010000) character is set up, in the second of the three control characters, which follow the block. If parity is to be checked, these three characters are read in, and checked as follows:-

```
          F    A    B
          0 . 17 . B0  .       .   . 1
          1 . 21 . B16 .       . 3 . 1
          2 .  6 . B17 . B17
          3 . 12 . 10  . R -            ——> error routine
               :
constant 17 .  4 .  2                      D10
```

The first instruction reads in the data block, but does not give a D18. The second instruction reads the three control characters into B17, again without a D18 as the buffer is automatically switched on being emptied. If parity has failed, B17 will contain a D10; if it has succeeded, the D10 position will be zero.

7.2.6. Other input/output functions

It is possible to use normal input/output functions 19 - 24 and 28, if desired, with tapes.

For example:

22 . B0 .   . 3 . 3

24 . B16 . 1 . 5 . 3

The first instruction will take the contents of B0, convert them to the decimal equivalent, and then transfer the last 3 decimal characters into the buffer. The second instruction will take the least significant 5 characters from B17 without conversion, transfer them to the buffer, and them switch to the next ½ buffer. On the tape, these will be 3 decimal characters, 5 alpha/numeric characters, and 3 control characters. The remainder of the 99 characters associated with the block transfer will not be used.

When reading back a tape prepared by this method, the equivalent input function must be used to convert the data back to the binary equivalent.

7.3. Punched Card Reader

7.3.1. Elliott Reader (Diagram 1)

7.3.1.1. Cards

Powers 65 column, Hollerith or I.B.M. 80 column with or without column recognition marks. Type to be specified with order.

7.3.1.2. Speed

400 cards per minute.

7.3.1.3. Controls provided on the reader

Mains Switch
Single Shot/Normal Switch
Buffer Reset Switch
Start Button
Stop Button.

7.3.2. Punched Card Codes

Information is coded on punched cards, one or more holes in one column representing a character. Automatic translation into the 6-bit computer code is effected by circuitry designed for the punched card code to be used.

7.3.2.1. I.C.T. Punched Card Code

Numbering the hole positions of the card from top to bottom:-

A B C 1 2 3 4 5 6 7 8 9

		A to I	Positions	A and 1 to A and 9
		J to R	"	B and 1 to B and 9
		S to Z	"	C and 1 to C and 8
Non column recognition	-	No character	"	C and 9
Column recognition	-	Space	"	C and 9
		1 to 9	"	1 to 9
		10	"	A
		11	"	B
		0	"	C

½	Positions	3 and 1
/	"	3 and 2
%	"	3 and 4
1/3	"	3 and 5
X	"	3 and 6
&	"	3 and 7
.	"	3 and 8
-	"	3 and 9

By special arrangement combinations may be made available for:

* ¼ ¾ ' ( ) + £ $ [redacted] 12

7.3.3. <u>Punched Card Layout - Diagram 2</u>

Information to be punched is arranged in 'fields', a group of such fields to contain relevant information on each item to be dealt with. When an item occupies less than half the available columns on a card, two or more items can be punched on each card. For ease of visual inspection cards may be printed with vertical lines separating fields and the punched card interpreted, the contents of each column being printed on the card.

The column recognition facility can be provided if required. This eliminates the necessity to punch every column on the card between the first and last columns containing information. For example, if two adjacent fields were allotted on the card to a twelve column name and a seven column number, then if the name to be punched is SMITH, the twelve columns will be punched SMITH NC NC NC NC NC NC NC i.e. seven 'no character' symbols will be punched after the name to fill the field if the column recognition facility is not available. Failure to punch the NC symbols would lead to SMITH plus the seven characters punched in the 'number' field being taken by the computer when twelve name characters are called for.

Column recognition cards are printed with a black rectangle corresponding to each column along one edge of a card. Where this facility is used, the card reader scans the edge printed with the markings, reading the character associated with each mark and inserting NC symbols automatically where no punching is found.

7.3.4. Card Reading

The maximum speed of the card reader is 400 cards per minute. To achieve this rate the information on each card must be taken by the computer in not more than 150 milliseconds. The card feed however is clutch driven, not simply cyclical, so that if the 150 milliseconds are exceeded only a proportional time is lost, not a complete cycle.

The buffer associated with the card reader consists of two 99 character matrices.

When cards are loaded into the card reader hopper and the start button is pressed, if the associated buffer is clear, the first card is fed to the reading platform. On pressing the button a second time the contents of the first card are read into the first 99 character matrix in 6-bit code, the matrices automatically switched to receive the second card into the second matrix, the third card remaining on the reading platform.

The contents of the first matrix are now available to the computer. When this information has been read the matrices will be switched, the second matrix being available to the computer and the third card read into the first matrix. This process will continue until all cards have been read.

Since a punched card contains a maximum of 65 or 80 characters depending on the type, it is arranged that the trailing end of the card fills the remaining character positions of the half-buffer with 'no characters'.

7.3.5. Card Input Programming

The information is called from the card reader buffer by the program, using functions 17, 19, 20, 21, block input, decimal, sterling and alphanumeric respectively.

The form of the instruction is similar for each function.

The address of the destination register is written in the 'a' address of the instruction.

The 'b' address is used to indicate that all required information has been taken from one matrix and for the computer to switch to the other matrix. This is accomplished by a 1 in the 'b' address or D18 position of the instruction calling for the last information required from a buffer i.e. card.

The 'C1' address is used, with the exception of function 17, to specify the number of characters or columns called for and the C2 to indicate the unit number of the card reader.

7.3.5.1. Function 17 - Block Input

This function calls into the register specified and the succeeding 15 registers, 96 characters from a matrix, with no further decoding from the 6-bit alphanumeric code.

The first character is stored in the most significant position of the register specified and the ninety-sixth in least significant position of the sixteenth word. The highest numbered characters will be 'no characters' generated on the sensing of the trailing end of the card.

A D18 must be given with this instruction to switch matrices.

The C1 address is redundant.

7.3.5.2. Function 19 - Input Decimal

The number of characters specified in the 'C1' address are decoded into a binary value and stored in the register specified in the 'a' address.

Since the highest positive decimal value that can be stored in 36 binary digits is 34,359,738,367, the maximum number of decimal characters which can be called for is 11. Where a negative number is to be used a minus sign can be included to make the maximum number of characters 12.

Up to 15 characters may be specified but where, on conversion the binary value exceeds 36 bits, the most significant bits are lost.

Where 0 characters are specified, one character will be transferred to the destination register.

7.3.5.3. Function 20 - Input Sterling

The number of characters specified in the 'C1' address are decoded into binary pence or halfpence (depending on machine specification) and stored in the register specified in the 'a' address.

The highest positive sterling value that can be stored is £143,165,576 10s. 7d binary pence or half this sum on a binary halfpenny machine. The maximum number of characters that can be stored is 12 or where a minus sign is included 13.

As with function 19, 0-15 characters may be specified with similar results.

7.3.5.4. <u>Function 21 - Input Alphanumeric</u>

Two registers are filled using this function regardless of the number of characters called for. The registers affected are the one specified and the next highest numbered register. In reading alphanumeric characters no decoding is carried out as the characters in the matrix are already in 6-bit alphanumeric code.

When less than twelve characters are called for, they are stored, right justified from the higher numbered register. For example, where characters next available from the matrix are SMITH, and the following instruction is performed:-

100	21		40			5	7	
101								

the result is :-

NC	NC	NC	NC	NC	NC	NC	S	M	I	T	H

Reg. 40 Reg. 41

It can be seen that NC symbols are inserted automatically in 'unused' character positions. This should be remembered when tests are performed on information read in using function 21.

Where 0 characters are specified 12 characters are transferred. When more than 12 characters are specified - 0 characters are transferred.

7.3.5.5. **Examples**

In the examples given the cards are punched with the layout shown in diagram 2.

Reading the contents of one card placing the contents into registers 400 - 403 inclusive.

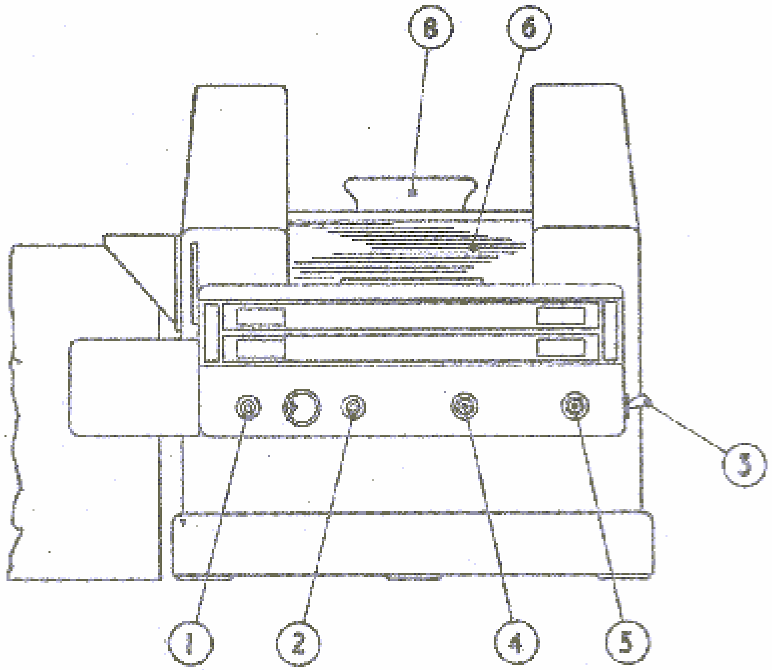
	F		A		B	C1	C2	M
32	21		400			12	7	
33	19		402			5	7	
34	20		403		1	7	7	

The instruction in register 34 switches the card reader buffer after 24 columns have been read.

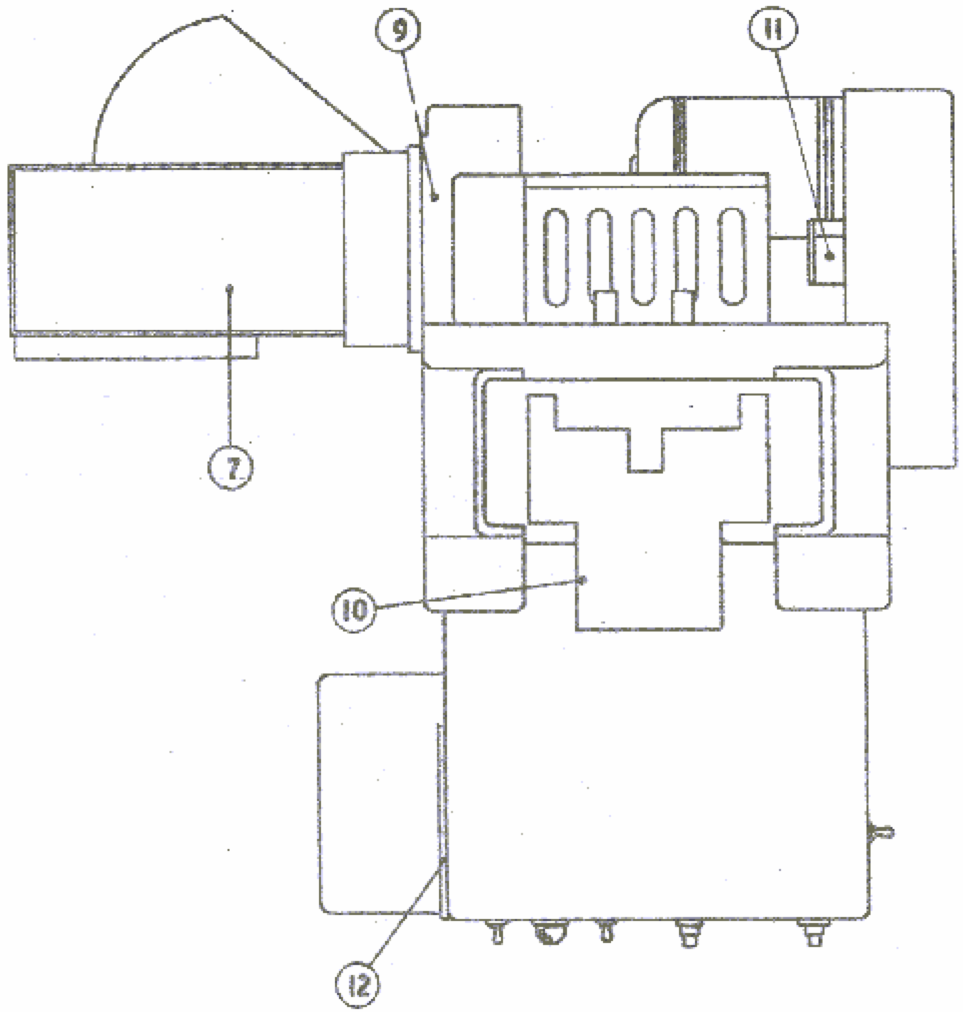
It is assumed in this example that the card is punched with three sets of fields to be read into registers 400-403, 404-407, 408-411

	F		A		B	C1	C2	M
32	1		39		7			
33 →	21		412			12	7	7
34	19		414			5	7	7
35	20		415			7	7	7
36	25		40		33			→
37	19		0		1		7	
38	13		0					→
39	0		-12					
40	0		- 4					

The instruction in register 37 switches the card reader buffer after 3 x 24 columns have been read.



FRONT VIEW



TOP VIEW

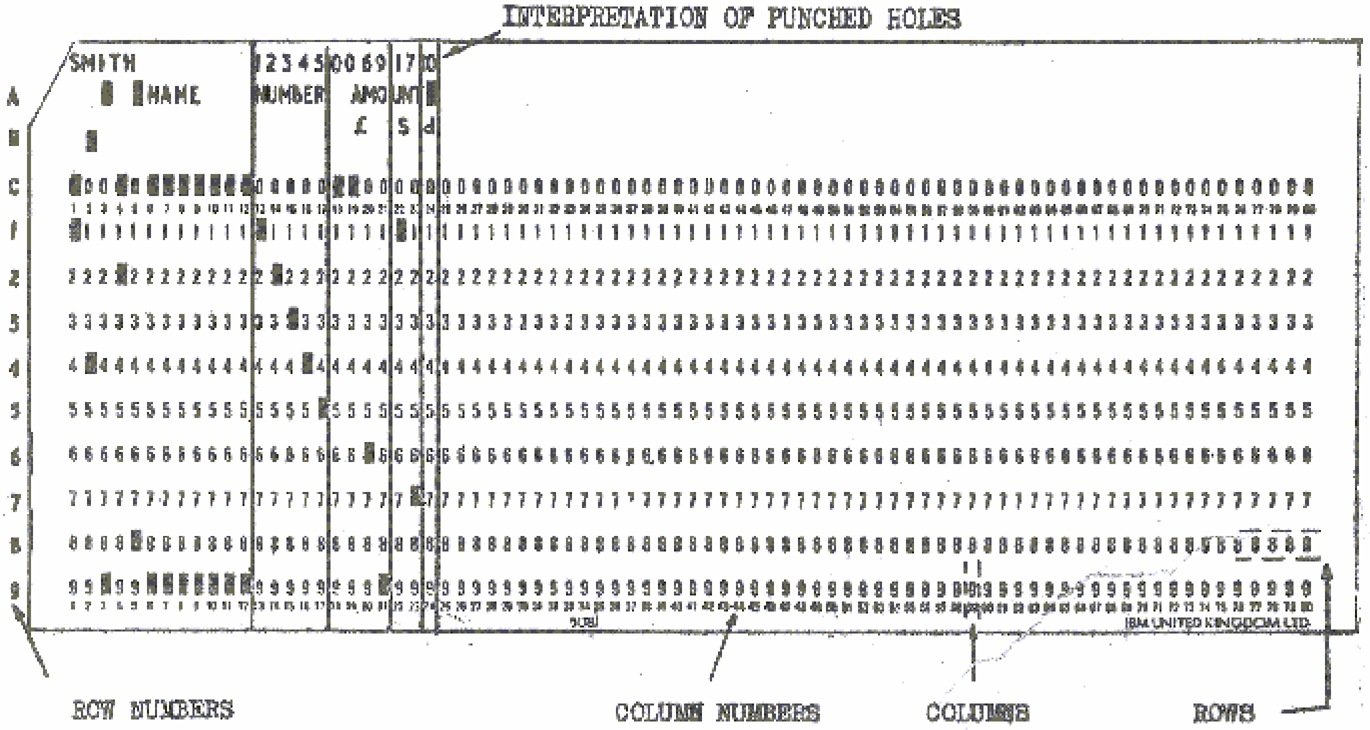
1 ON/OFF SWITCH
2 SINGLE SHOT/NORMAL SWITCH
3 BUFFER RESET SWITCH
4 START BUTTON
5 STOP BUTTON
6 FEED HOPPER
7 CARD RECEIVER
8 PRESSURE PLATE
9 READING HEAD
10 PICKER KNIFE
11 FEEDING ARM
12 CARD ALIGNMENT PLATE

DIAGRAM 1

CARD READER

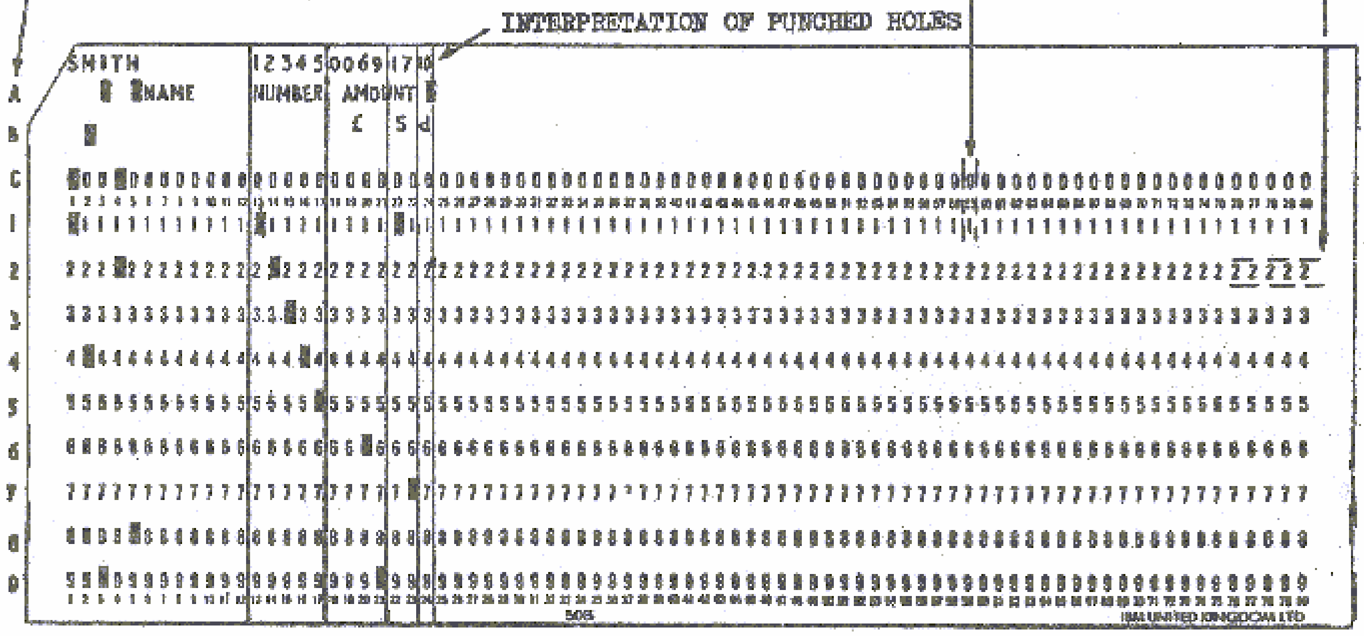
SPECIMEN CARD - HOLLERITH OR I.B.M. TYPE

PUNCHED WITHOUT COLUMN RECOGNITION FACILITY



SPECIMEN CARD - HOLLERITH OR I.B.M. TYPE

PUNCHED WITH COLUMN RECOGNITION FACILITY



N.B. COLUMN RECOGNITION MARKS ARE ON THE REVERSE SIDE OF THE CARD

DIAGRAM 2

7.4. Paper Tape Reader

7.4.1. Ferranti Reader, TR5 (Diagram 1)

Channels: 5, 6, 7 or 8 hole tape.

Speed: 300 characters/second.

7.4.2. Controls provided on the reader

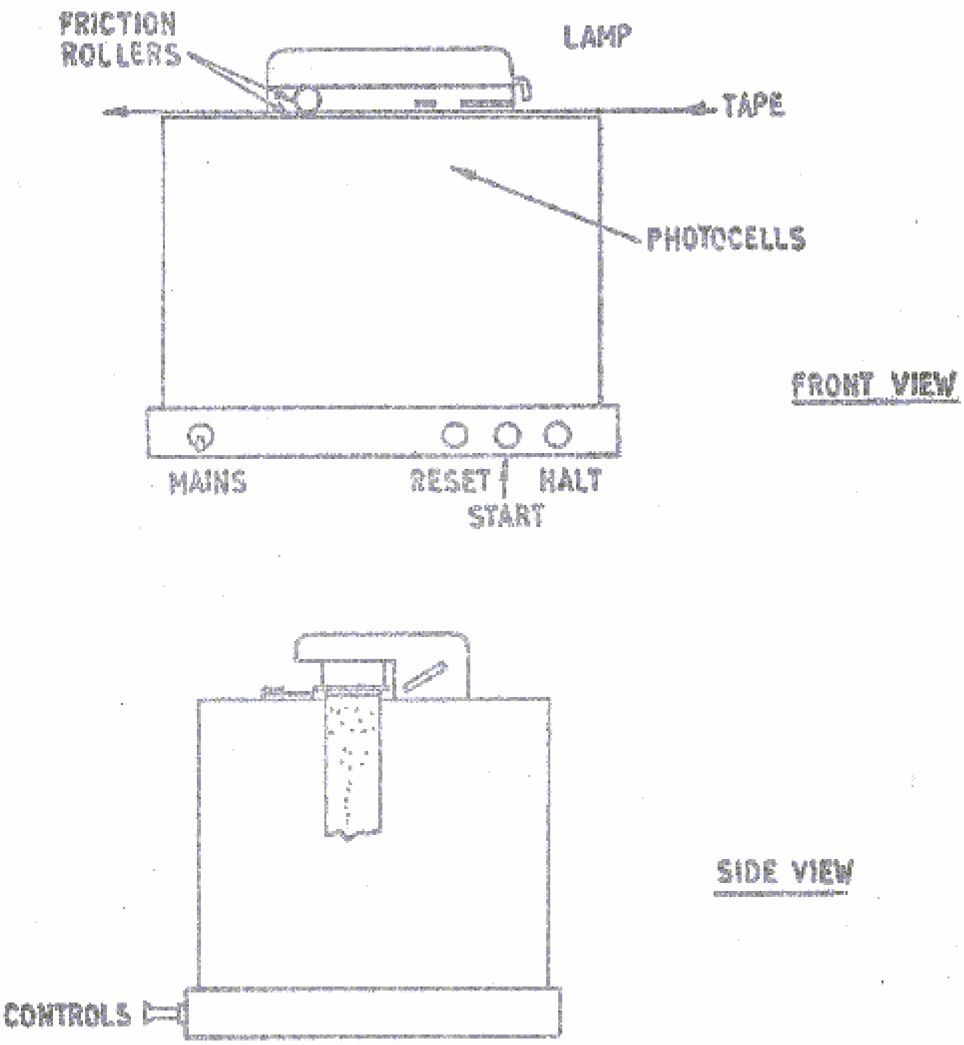
Mains switch

Reset : stops tape and clears buffer (Sec. 6.1).

Halt: stops tape, but leaves buffer unaffected.

Start: initiates process of reading tape into next available buffer matrix (Sec. 4).

FERRANTI PAPER TAPE READER TR5 DIAGRAM 1



7.4.3. Paper Tape Codes

1. Information is coded on paper tape in a 5, 6, 7 or 8 hole code. This code does not necessarily correspond in any way with the computer 6-bit code; automatic translation is provided. A 5-hole code used at I.C.T. is given below as an example (diagram 2).

2. The paper tape code includes, besides information characters such as A, B, C, 1, 2, £, etc., a number of special characters. These may vary with different codes, but will normally include:

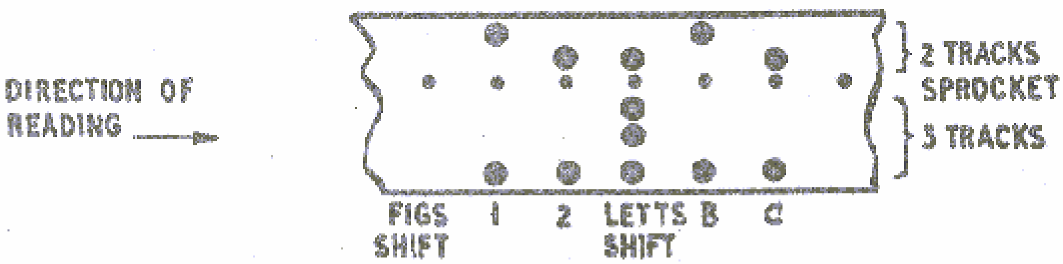
Number End, signifying the end of a variable-length group of information characters on the tape.

Figures Shift and Letters Shift (5-hole codes only), indicating that the following information characters are to be interpreted as being in figures or letters case.

Erase (all holes), used to obliterate any unwanted character on the tape.

Run Out, used at the beginning and end of a tape to allow for handling, etc. (With a 5-hole code, Figures or Letters Shift is normally used for this purpose). (See 6.2).

I.C.T. 5 TRACK PAPER TAPE CODE — DIAGRAM 2



INTERPRETATION OF A PUNCHING DEPENDS ON PRECEDING SHIFT CHARACTER.

PUNCHING	VALUE	FIGURES CASE	LETTERS CASE
000 . 00	0	Figures Shift/Run Out	
000 . 01	1	%	P
000 . 10	2	£	Q
000 . 11	3	– (Minus)	R
001 . 00	4	*	S
001 . 01	5	/	T
001 . 10	6	. (Stop)	V
001 . 11	7	¼	W
010 . 00	8	¾	X
010 . 01	9	' (Apostrophe)	Y
010 . 10	10	(	Z
010 . 11	11	)	D
011 . 00	12	+	M
011 . 01	13	x (Multiply)	U
011 . 10	14	½	&
011 . 11	15	Space	
100 . 00	16	0	A
100 . 01	17	1	B
100 . 10	18	2	C
100 . 11	19	3	E
101 . 00	20	4	F
101 . 01	21	5	G
101 . 10	22	6	H
101 . 11	23	7	I
110 . 00	24	8	J
110 . 01	25	9	K
110 . 10	26	10	L
110 . 11	27	11	N
111 . 00	28	12	O
111 . 01	29	Carriage Ret./Line Feed/No. End	
111 . 10	30	Letters Shift	
111 . 11	31	Erase	

7.4.4. Decoder and Buffer (Diagram 3)

1. The buffer consists of two matrices, each holding up to 12 characters in the 6-bit, internal code. The matrices are used alternately.

2. When loading a paper tape, the normal procedure is to Reset (clearing the buffer). Set the tape in the reader and Start.

3. Characters are now read from the tape and inspected. Normal information characters are automatically translated into the 6-bit internal code and stored in the buffer (Sec. 6.3).

4. Any special characters are not passed to the buffer, but are treated differently.

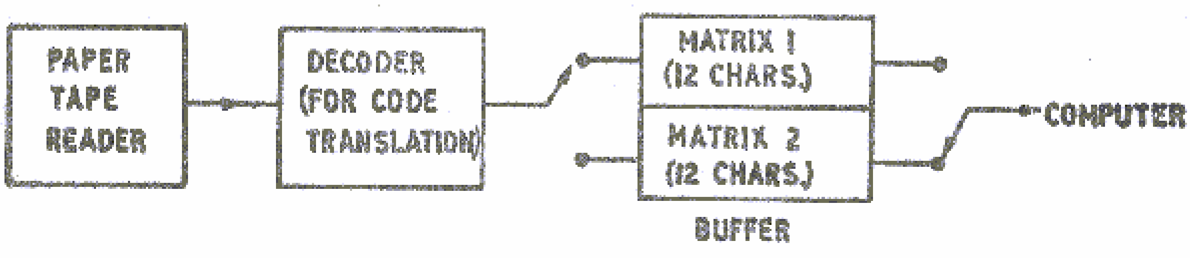
   Erase and Run Out are ignored.

   Figures and Letters Shifts are used in the translation of following information characters.

   Number End is used to switch the input to the next matrix. Thus a group of up to 12 information characters followed by Number End will be stored in one matrix.

5. The input will also be switched to the next matrix when one matrix contains 12 information characters.

PAPER TAPE READER BUFFER DIAGRAM 3



7.4.5. Reading from the Buffer

1. Input functions are

   Fu. 19 : input with decimal conversion.

Fu. 20: input with sterling conversion.

Fu. 21: alphanumeric input

(See Section 7.1 - Input and Output Instructions).

2. When a number followed by Number End has been read into the buffer from the tape, the digits of the number are stored in one buffer matrix, and a marker is left indicating the next free place in the matrix (see diagram 4)

3. To read this number, an input instruction calling for 12 characters is used. This takes first the unfilled positions in the matrix (which appear as 'no characters') and then the digits of the number in the order in which they were read from the tape.

4. A D18 (1 in the B address) must also be given, to switch the buffer output to the next matrix and to return the marker to the first position in preparation for the next reading operation from the tape.

5. Example. Tape punched with the following characters: 3, 2, 1, 4, 5, Number End.

Buffer:

3	2	1	4	5							

↑ (marker)

Input instruction 19 . 50 . 1 . 12 . 11 places binary equivalent of 32145 in register 50 (Channel 11 is tape reader).

Input instruction 20 . 51 . 1 . 12 . 11 places binary equivalent of £32. 14. 5 in register 51.

Input instruction 21 . 52 . 1 . 12 . 11 places

NC	NC	NC	NC	NC	NC	NC	3	2	1	4	5

in registers 52, 53.

6. The contents of the buffer matrix may be read by two (or more) instructions. In this case the instructions call for 12 characters altogether, and only the last contains a D18 (Sec. 6.4 and 6.5).

7. Example. Tape punched with quantity (variable length) and code (4 characters) thus:

144 ABCD Number End.

Buffer:

1	4	4	A	B	C	D					

Input instruction 19 . A1 . 0 . 8 . 11 places the binary equivalent of 144 in register A1 and leaves the buffer thus:

✓	✓	✓	A	B	C	D	✓	✓	✓	✓	✓

Second input instruction 21 . A2 . 1 . 4 . 11 places

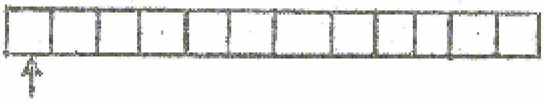
NC	NC	NC	NC	NC	NC	NC	NC	A	B	C	D

in A2 and A3; the D18 resets the marker.

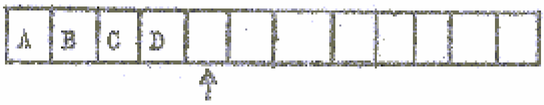
OPERATION OF BUFFER

DIAGRAM 4

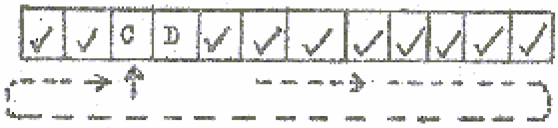
Matrix empty; marker set at first position.



When filling buffer, marker shows next empty position.



When reading, any unfilled positions are read first, then the characters taken from tape.



After emptying, D18 returns marker to first position.

7.4.6. Notes

1. If the Reset button is pressed while the computer is held up on an instruction addressing the paper tape reader, it is necessary to repeat the instruction, as the stored 'request' signal has been destroyed.

2. In addition to the characters used for computer input, the code will probably contain some characters required when tapes are to be printed on a teleprinter or electric typewriter. For example the I.C.T. code shown uses the

Number End symbol as Carriage Return/Line Feed signal also, and a Tab character is also found in some codes.

3. Some print control characters (e.g. Tab) may be treated by the computer input as information characters.

4. If two or more pieces of information (e.g. code and quantity) precede a Number End, only the first may be of variable length.

5. If more than 12 characters are called from a matrix before the D18 is given, the extras are zeros.

6. The conventional channel numbers for paper tape readers are 11 and 12.

7.4A The Elliott High-Speed Paper-Tape Reader

7.4A.1 GENERAL DESCRIPTION

The Elliott High-Speed Paper-Tape Reader operates at any speed up to 1000 characters per second. It can read 5-, 7- and 8-hole punched paper tape, as well as 6-hole square-punched Olivetti tape.

The instrument stands on a plinth which contains data control and other electronic circuitry, and which houses the manual controls that operate the machine.

7.4A.2 CONTROLS SITUATED ON THE READER UNIT

(a) The Load Bar The load bar extends along the front edge of the reading platform. It has two separate movements as follows:

i. The bar can be pulled outwards, in a horizontal plane away from the platform, in order that the reader can accommodate tape of various widths. In its horizontal travel, the bar sets itself naturally into 4 different positions, permitting the use of 5-hole tape (innermost position), 6-hole Olivetti tape, 7-hole tape and 8-hole tape (outermost position).

ii. The 3-position vertical movement provides the following facilities:

A. (Rest Position) Reader Functions Normally

B. (First Depressed Position) Arrests tape movement and resets the control circuitry.

C. (Final Depressed Position) Releases the Paper Break to permit a replacement of tape.

(b) The Start Button Initiates the reading of data into the next available control matrix.

(c) The On-Off Switch Applies power to the tape tractor motor and to the photo-cell luminary.

7.4A.3 CONTROLS SITUATED ON THE PLINTH

(a) Mains On/Off Controls the main power supply to the entire equipment.

Reset Arrests the tape and clears the [illegible]

Halt Stops the tape but leaves the buffer unaffected.

Select Paper-Tape Switch This rotary switch has two positions, one for Olivetti and the other for round-hole tapes. On some earlier models, however, the switch has four positions, marked: "Olivetti", "5-hole", "7-hole" and "8-hole". In either case, the switch has to be set to correspond with the tape being used.

When it is required to change to read-in a different form tape, an engineer should be consulted to determine whether a [c]ard-change in the plinth is necessary.

[N].B. The controls at the rear of the plinth (fuses, lamp - and marginal-checking devices) are for the use of engineers.

THE INDICATOR LAMPS

(a) Test (Red cover) Shows that the instrument is being marginally checked for reliability by engineers. This lamp should normally be off.

(b) Parity Failure (Blue cover) Indicates a coding error.

(c) Halted (Red cover) Lights up when the halt or reset controls have been operated, and when the load bar has been depressed.

(d) Mains (Amber cover) Indicates that power is available to the equipment and that the mainswitch is on.

.5 Loading the Tape

(a) Set the lateral position of the load bar and the paper-tape type selector switch on the plinth to suit the width of tape to be used. If a new setting is necessary, enquire of an engineer that all else is in order for reading the new tape, as in 7.4A.3(d) above.

(b) Depress the load-bar sufficiently to allow the tape to slide under the guiding and driving assembly. Load the tape so that there are 3 rows of holes between the sprocket perforations and the reading cabinet, or, if Olivetti tape is being read, ensure that the guide (i.e. the wider) edge of the tape is facing away from the operator.

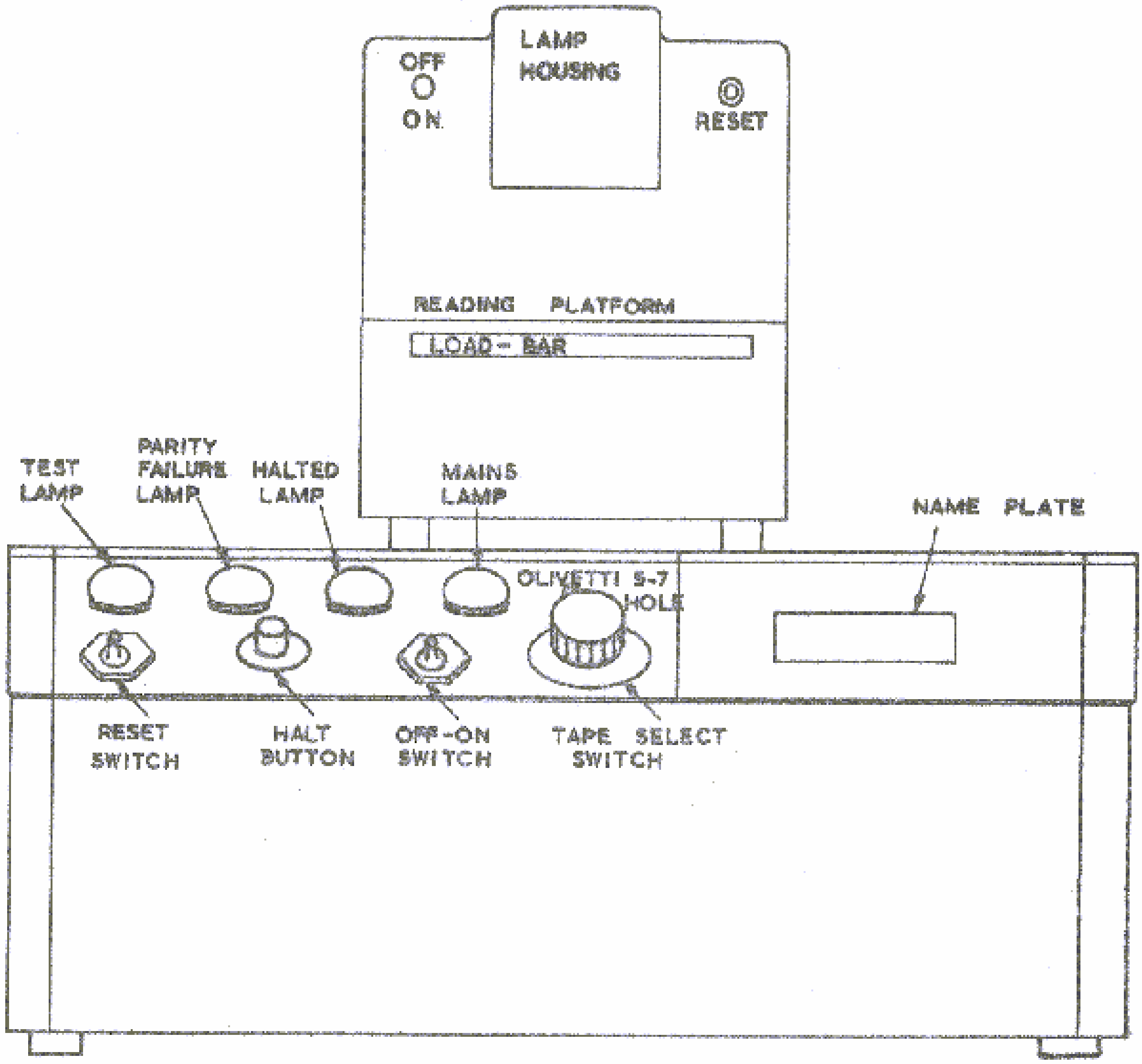
(c) Ensure that the tape is in its correct running position before starting the machine. This position is attained when the rear edge of the tape is just touching the guide-stops at the back of the reading platform.

(d) Release the load bar, and operate the reset switch to clear the buffer. The tape will commence being read when the "start" button is pressed.

7.4A.6 CODING

There are no differences in the methods of coding for programmers, no matter what type of tape is to be employed. Data is correctly deciphered for the buffers when the selector switch is appropriately set and when engineers have installed suitable cards in the plinth.

NOTE: For maximum reliability, the manufacturers recommend that no spool used in conjunction with the High Speed Reader should have a diameter greater than 5 inches.



OUTLINE SKETCH H/S PAPER TAPE READER WITH PLINTH ILLUSTRATING THE PRINCIPAL OPERATORS CONTROLS

DIAGRAM 1.

TL. 1063/7.4A

7.5. Line Printer - Samastronic

7.5.1. Equipment:- Powers Samas Samastronic

7.5.1.1.	Speed -	300 lines per minute
7.5.1.2.	Line Length -	140 characters at 10 characters per inch.
7.5.1.3.	Paper Movement :-	2 webs of stationery, independently controlled. Minimum width - 3 in. Maximum width - $17\frac{3}{4}$ in. 6 or 8 lines per inch vertical spacing, decided at time of ordering. Multiple line feeding on either web up to a maximum of 1 inch. Long feeding is available on Web 1 only. Maximum long feed - 16 in.

Characters:-

The range of characters available on the Samastronic printer is:-

0 to 11
A to Z
$\frac{1}{4}$ $\frac{1}{2}$ $\frac{3}{4}$ + - (Minus) * (asterisk)

% £ .(stop) / $ &(ampersand) '(apostrophe)

▓ (guillotine control) 1/3 x ( ) 12

All 57 symbols are available.

The printer leaves a blank where it is sent a signal for either 'space' or 'no character'.

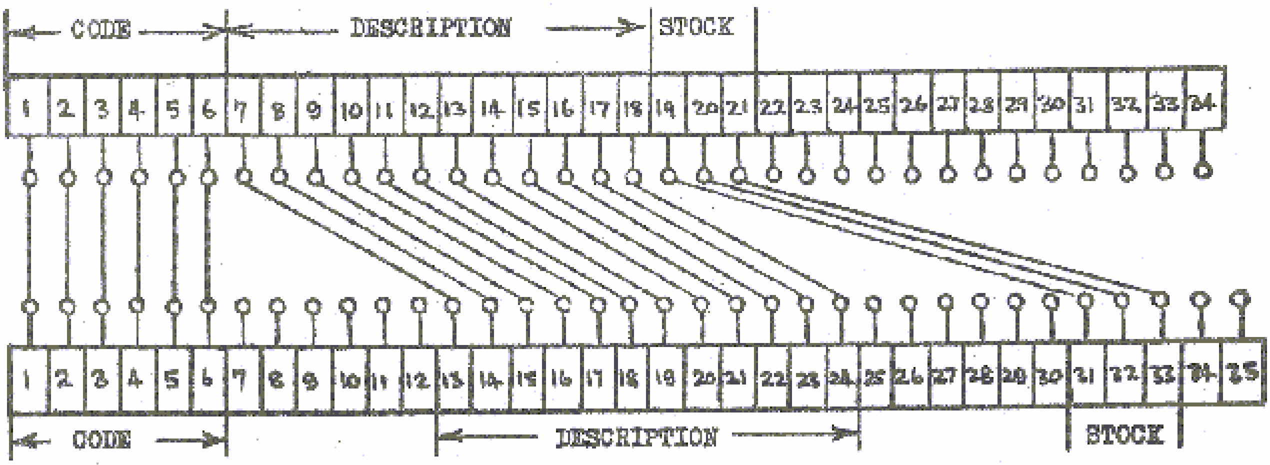
Characters sent to the printer buffer which are not included in the character range for the printer will also have the same effect as 'space' and 'no character'.

7.5.2 Operation of Printer and Buffer

It will be seen that information from the computer is put out into the buffer, which has two matrices of 99 characters each. From the buffer it is called by the printer into the Print Line Store.

The Print Line Store has a capacity of 140 character positions, and the characters set up here are decoded and the results used to print the appropriate symbols. The decoded outputs are taken to the print head through an Electrical Connection Unit or E.C.U., which allows any of the character positions in the Print Line Store to drive any print position on the print head. There is a choice of five E.C.U.'s available during any one job, selected by program instructions, and the E.C.U.'s may be changed between jobs. A particular layout of printing on the line may therefore be achieved either by using a 'one-to-one' E.C.U. (i.e. one which connects the P.L.S. outputs directly to the corresponding print positions), and putting out the information in the required order and with appropriate spaces between each item. Or alternatively the information may be put out in the order most convenient to the program, and an E.C.U. used which arranges it correctly on the line. In practice, a combination of the two methods is used, with special E.C.U.'s provided for the most frequently printed lines.

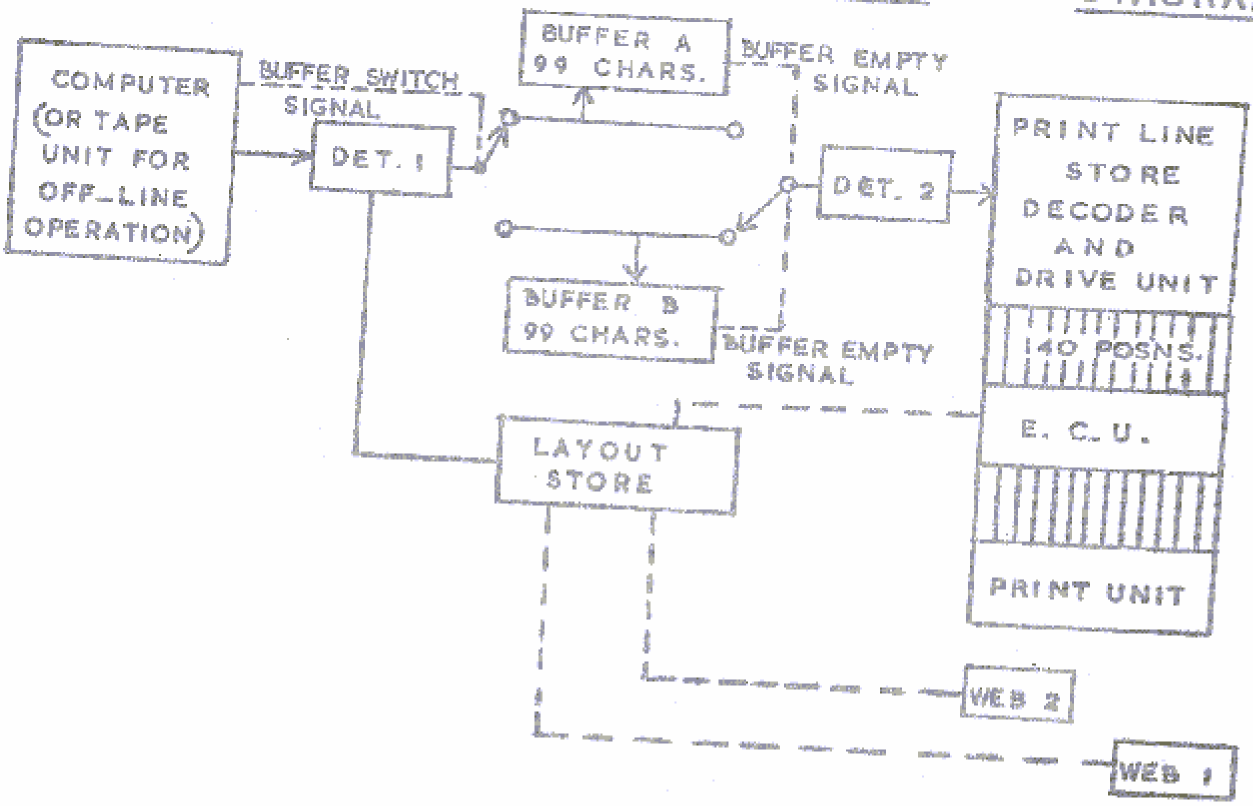
As an example of the use of the E.C.U.; consider the printing of a stock list showing stock code number, a brief description, and quantity in stock. On the printed list, the code number occupies 6 positions, there are then 6 spaces and the 12 - character description, 6 more spaces and a quantity of up to 3 digits. The program has put out to the Print Line Store (via the buffer) the 6 - digit code number, 12 - character description, and 3 digit stock quantity. Then an E.C.U. to place these in the correct positions on the line would be connected thus:



It should be noted that the E.C.U. may be used to connect any Print Line Store position to two (though not more than two) positions on the pr[illegible] line, thus duplicating the information. Thus in the example above, if a second list showing code number and stock only were required, this could be provided on the second web by connecting the corresponding E.C.U. inputs to two outputs. Or again, if selected items only were required on the second list, two E.C.U.'s could be used; one of which gave the simple list, and the other the duplicate. The program could then call for whichever was appropriate.

DIAGRAM OF PRINTER AND BUFFERS

DIAGRAM



7.5.3. Output Instructions

Function 22 - Output decimal
23 - Output sterling
24 - Output alphanumeric
28 - Output spaced sterling
18 - Block output.

7.5.3.1. Form of Instruction

Fn	A	B	C1	C2	M
	Source Register(s)	Printer Control	No. of Char-acters	Unit No.	

Output instructions involving a conversion from binary (i.e.functions 22, 23 and 28) are first converted in the Arithmetic Unit giving a result 12 characters long. Non-significant zeros to the left of the first significant character are then converted to 'spaces'. If all characters are zero characters then all but the least significant zero are converted to 'spaces'.

Alphanumeric output (function 24) addresses a pair of registers, viz. the register specified and the following one, and again 12 characters are available in the Arithmetic Unit. Zero-suppression is not performed.

Up to 12 characters may be specified in the C1 address. If less than 12 characters are specified they are counted from the less significant end of the characters held in the Arithmetic Unit.

7.5.3.2. Example 1

22		100			3	9	

Register 100 contains, in binary, the number 5,000,123. This number, converted to decimal, will first appear in the Arithmetic Unit as follows:-

0	0	0	0	0	5	0	0	0	1	2	3	ARITHMETIC UNIT

The non-significant zeros are then suppressed automatically giving this result

Space	Space	Space	Space	Space	5	0	0	0	1	2	3	ARITHMETIC UNIT

The number of characters specified, in this case 3, are then transferred to one of the buffers thus:-

1	2	3					…					99 CHARACTER BUFFER

If 5 characters had been specified the result in the buffer would have been thus:-

0	0	1	2	3			…					99 CHARACTER BUFFER

If 11 characters had been specified the result would have been thus:-

Space	Space	Space	Space	5	0	0	0	1	2	3	…		99 CHARACTER BUFFER

7.5.3.3. Function 28 (output spaced sterling) inserts a space character between pounds and shillings and a space character between shillings and pence. These two spaces must be included in the number of characters specified to be sent to the buffer. Where a machine is arranged for binary halfpence, function 28 is not available.

7.5.3.4. Example 2. Spaced Sterling Output

Instruction	28		100			6	9	

Register 100 contains, in binary, the number 270.

When this instruction is performed, the characters transferred to the buffer will be:-

1	Sp	Sp	2	Sp	6				…	

It will be seen that, when the digit which represents tens of shillings is zero, it is converted to 'space', giving zero suppression when printing.

7.5.4. Control Information and Buffer Switching

The 'b' address of the output instruction to the printer is used to switch half buffers and to provide control information for printing purposes. When the 'b' address of an output instruction is zero, no printing will take place, information merely being transferred to the buffer.

The ten bits of the 'b' address are used viz:

D18	To switch half buffers
D19-21	To specify the E.C.U. through which the line of information is to be printed.
D22-24	To specify the number of line feeds for the Web 2 stationery.
D25-27	To specify the number of line feeds (or a long throw) for the Web 1 stationery.

7.5.4.1. E.C.U. Digits 19-21

During any one operational run, up to five E.C.U.'s are available. These are numbered 1 to 5 and the binary value of digits 19-21 is used to select the appropriate E.C.U. through which the line of information is to be printed. Using digits 19-21 it is also possible to specify E.C.U.6 and E.C.U.7 and these are used for off-line printing purposes (See 7.5.6.)

7.5.4.2. Paper Movement Digits 22-24 and 25-27

The paper movement signals are shown in the following tables:-

7.5.4.3.

Value of the 3 bit D22-24 or D25-27	Significance (for an eight lines per inch printer).
0	No movement
1	Single line spacing
2,3,4	2, 3 or 4 line spacing
5	6 line spacing
6	8 line spacing
7	long feed (digits 25-27, Web 1 only)

Line spacings or long feed take place after the line with which they are given has been printed.

With a six line per inch printer values 5 and 6 give 5 line and 6 line spacing.

Both webs of paper may be fed at once if desired.

7.5.4.4. Long Feeding

As the above table shows, when digits 25-27 of the output instruction contain a value of 7, a long feed or long throw occurs after printing.

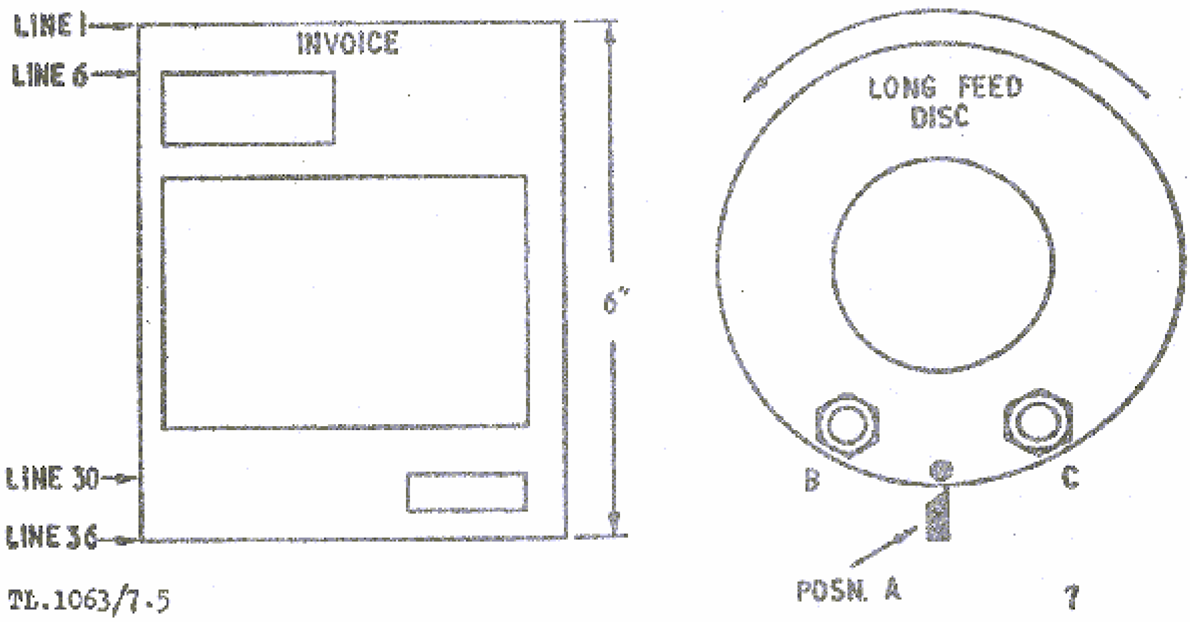
The movement of the stationery is controlled by a long feed disc the circumference of which corresponds to the form being printed. As the stationery form moves under the print heads the disc revolves through a corresponding distance, i.e. stationery and disc move in step.

The disc has one or more stopping studs which are positioned to correspond to the points on the form at which it is desired to print after long feeding.

When a long feed paper movement signal is given a stop arm is activated, the paper feeds and disc revolves, until the stop arm makes contact with the next stud.

For example, an invoice often contains a variable number of printed lines within the body of the form and has a total printed in a fixed position at the foot. In addition the invoice name and address is also printed in a fixed position at the head. When printing this form, a disc with two stopping studs would be used.

DIAGRAM 2



In Diagram 2 position A on the long feed disc corresponds with line 1 on the invoice form; the stopping stud B with line 6 and C with line 30.

At the start of a normal printing job, the paper would be positioned with line 1 under the printer heads and the long feed disc with position A opposite the stop arm.

The program will first throw the paper so that line 6 is in a position ready for printing, (i.e. stud B making contact with the stop arm at position A). The name and address and body of the invoice will now be printed using normal line feeds with the exception of the last line of the body. This will be printed using an instruction containing long throw digits.

The paper at this point is positioned so that line 30 (i.e. the total), can be printed and stopping stud C will be at position A. As the total is printed a long throw signal is again given and the paper will move forward to line 6 on the next invoice form.

N.B. The following restrictions must be observed:-

1. Stopping studs cannot be less than 1 inch apart.
2. Throws of less than ½ inch must not be attempted i.e. a long throw signal must not be given when the paper has less than ½ inch to travel to the next stopping point.

7.5.4.5. <u>Example 3</u>

In example 2, a binary value was converted to spaced sterling, and output to the printer buffer. If it were required to print this information double spaced (i.e. 1 blank line following the printed line) on Web 2, using E.C.U. 3, the 'b' address of the output instruction would be as follows:-

WEB 1			WEB 2			E.C.U.			D18
0	0	0	0	1	0	0	1	1	1

The decimal equivalent of the 'b' address is 39 and this would be written in the 'b' address of the instruction viz:

28		100		39	6	9	

A non-zero 'b' address in an output instruction causes three extra characters to be sent to the buffer, immediately following the information transferred by the instruction. The first of these three characters is a special marker character (binary 36) and the second and third are characters formed by adding two zeros to the significant end of the ten bits which comprise the 'b' address, making 12 bits in all, or two 6 bit characters.

7.5.4.6. <u>Example 4</u>

When the instruction shown in example 3 is performed, the information and marker characters sent to the buffer will be:-

1	Sp	Sp	2	Sp	6	36	0	39		

Referring to the Printer and Buffer Diagram 1 (see 7.5.2), when detector 1 recognises the marker character, the two following characters are tested for non-zero E.C.U. digits and if found, the information is stored in the layout store. Whether the E.C.U. digits are zero or non-zero, the three special characters are always written to the buffer.

When information is transferred from the half buffer to the print line store, that is, when the half buffer is switched, detector 2 will detect the marker character and look for non-zero E.C.U. digits which signify the end of the print line. When these are found, using the control information available in the layout store, the information in the print line store is printed through the specified E.C.U. and the paper on either or both webs is moved the specified number of lines after printing. Detector 2 does not allow the three special characters to be transferred to the print line store.

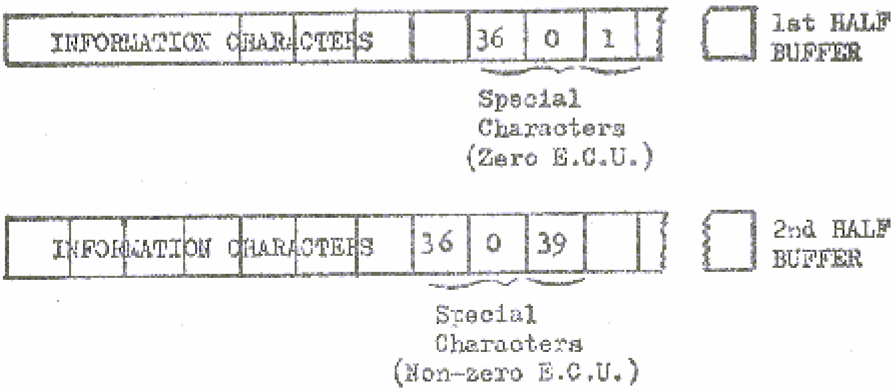
Each print cycle, that is, every one fifth of a second, printing will take place provided both of the following conditions are satisfied:-

1. Information is available in the print line store (i.e. the buffer has been switched)
2. A layout is available with non-zero E.C.U. digits.

Using this method it will be seen that there need not necessarily be a relationship between a line of print and one half buffer. Thus if a line of more than 96 characters is to be printed, both halves of the buffer will be required, this being achieved by giving a D18 when a suitable amount (N.B. not more than 93 characters) has been sent to the first half buffer. Because the last instruction contains a non-zero 'b' address (D18) the three special characters will follow the information into the buffer. The layout digits will be included in the instruction which sends the last characters of the line to be printed to the other half buffer.

When this instruction is performed, as previously described, the three special characters which include lay-outs will be placed immediately following the information in the second half buffer.

The two half buffers are shown below:-



When the information from the first half buffer is transferred to the print line store, detector 2 will detect and suppress the three special characters. Because the layout information is zero, this indicates that the line of print is not yet complete and the information is stored in the print line store, awaiting the balance of the line and layout information.

When the balance of information from the second half buffer is transferred to the print line store completing the line, the detector again suppresses the three special characters and, because the layouts are non-zero, printing will occur, using the layout information previously stored in the layout store. All the information in the print line store will be printed.

It is also possible to hold more than one print line in one half buffer by giving layouts without a D18 with the instruction at the end of each line. The instruction which contains the layout for the last line to be held in this half-buffer must also contain a D18 to effect buffer switching.

The layout store has a capacity of 5 layouts and therefore not more than 5 lines of print should be sent to two consecutive half buffers. Furthermore, a half buffer with four lines of print in it must not immediately follow a half buffer with one line of print in it.

Thus permissible arrangements of lines in successive half-buffers may be summarised as follows:-

1 line ; 1 line ; 1 line ; 1 line ; ..... (normal)
1 line ; 2 lines; 1 line ; 2 lines; .....
1 line ; 3 lines; 1 line ; 3 lines; .....
2 lines; 2 lines; 2 lines; 2 lines; .....
2 lines; 3 lines; 2 lines; 3 lines; .....
1 line ; 0 lines; 4 lines; 1 line ; 0 lines ; 4 lines; ...

Other arrangements are possible, but these should be sufficient for all normal purposes.

When considering how many lines of print can be held in any half buffer, the three special characters which follow each line must be taken into account.

Half buffers may be switched by the program either by giving a D18, or by filling all 99 positions with information and layout characters. However, a D18 must not be given when the half buffer has more than 93 characters in it already. If possible, this particular circumstance should be avoided; if it arises, the switch may be performed by filling the remaining places with 'space' characters or layout characters (the layout having digits in D19-27 only). Care must be taken that the characters output by any one instruction do not straddle the two halves of the buffer.

7.5.4.7. Example 5

If say, 94 characters had been previously output to the buffer the situation would be thus:-

	91	92	93	94	95	96	97	98	99
I	NFORM	ATION	CH	ARS.					

HALF BUFFER

Under these circumstances it is necessary to output two 'space' characters and for the instruction to contain layout digits but not a D18.

For example:-

24		105		38	2	9		

Register 106 contains 'space' characters.

The buffer will then appear thus:-

	91	92	93	94	95	96	97	98	99
INFORMA	TION	CHA	RS.		Sp.	Sp.	36	0	38

and being completely full will be switched.

7.5.4.8. Notes

When an instruction sends out characters which straddle two half buffers, during the switching from the first to the second half buffer at least three characters will be lost, and the following results will occur:-

1. When the instruction includes a layout, then:-

   a) If the three special characters are not included in those which are lost, i.e. if they are transferred to the second half buffer, the line will be printed correctly but three or more characters will be missing.

   b) If the first of the three special characters is lost (i.e. the marker character, value 36) then:-

      If a further instruction, sending information to the second half buffer, includes a layout with non-zero E.C.U. the information in the first half buffer, together with the information sent to the second half buffer (but no more than 140 characters which may include one or two of the special characters if they were not lost) will be printed. The layout used for printing will be that sent with the instruction which straddled the half buffers. Subsequently, all information printed will be one layout out of step.

2. When the instruction does not include a layout the results will be as described in 1(b) but in this case the correct layout will be used.

When a D18 is given after more than 93 characters are already in the half buffer the effects described in the preceding notes may occur. That is to say, information will be lost, lines will be printed with incorrect layouts and spurious information may be printed.

Layouts

Under normal conditions each time a layout is included in an output instruction, a layout is stored in the layout store and the corresponding special characters are stored in the buffer.

If for any reason the special characters are lost from the buffer (as described in the previous notes) then the lines of print in the buffers will appear to be out of step with the layouts in the layout store.

Similarly if the rules governing the use of the layout store are contravened, the same condition will apply.

7.5.5. Block Output to the Printer

A block output instruction (function 18) may be used with the printer, for example to put out a name and address. In this case, layout characters must be included among the information, if it is to be printed as more than one line. The limitation on the number of lines of print (four in any two consecutive half buffers is dictated by the size of the layout store) and must be borne in mind. The layout for the last line may be included in the B address of the output instruction, though, since 96 characters are transferred this must not contain a D18. Alternatively, a 'D18' may be included in the last layout included among the information, provided this

... act as a buffer switch ... characters 94-96. This will not ... buffer switch is achieved by putting a D22 only in the B addr... of the output instruction. Three characters are sent out (since the B address is non-zero), filling the half buffer, th... causing the switch, but they have no effect on the printer sin... the E.C.U. digits D19-21 are all zero.

When the printer calls for information from the buffer the characters are transferred a line at a time to the Print Line Store, the end of each line being recognised by the presence of the layout characters. When the last line has been taken, either the D18 in the dummy layout, or the fact that the end of the half buffer has now been reached, will cause the buffer switch to occur on the output side of the buffer. In the former case, any remaining characters making up the 96 character block will be lost.

7.5.6. Off-line Printing

The printer may be used separately from the computer to perform printing as an off-line operation. It is first necessary to record the information to be printed on magnetic tape and this is achieved by plugging a tape unit to the output channel called by the print instructions. This is done on the channel selector plugboard. When the tape has been recorded the tape unit buffer will be connected directly to the printer control circuitry bypassing the printer buffer.

When a tape is to be prepared, four extra instructions are needed in the program. The first, at the beginning of the job, is to rewind the tape unit and set it to write, the second is a dummy print instruction, after the last line of print, specifying E.C.U. 6 or 7 (see later) and the last is needed at the end of the job to rewind the deck and set it to finish. In accordance with normal practice when recording on tape, it is also necessary, before rewinding, to give an extra buffer switch to the tape unit buffer to ensure that the previous record instruction is properly carried out.

Normal printing instructions are used. The output functions 22, 23 and 28 will produce the usual conversion when addressed to a tape unit, so that the information on the tape is in converted form ready for printing. Function 24 will output the number of characters specified without conversion. The tape unit will record one tape block with each alternate D18. Every D18 will switch one quarter (i.e. 96 characters) buffer. Each print line is followed by three layout characters on the tape, so that, if, for example, two lines of print, each of 12 characters, and each with layout information and a D18, were output from the computer, they might appear on the tape as follows:-

Block Marker	Characters			
	1 to 12	13 to 15	16 to 99	100 to 111
	Data - 1st print line	marker and layout digits	not used	Data - 2nd print line
	112 to 114	115 to 198		
	marker and layout digits	not used		

Alternatively, the lines would appear as the second half of one block and the first half of the next, depending on whether the number of D18's given so far is odd or even.

It should be noted that a print instruction with layout information contains digits normally interpreted by a tape unit as a rewind instruction. The unit will, however, ignore these digits as long as there is an E.C.U. specified, which should, of course, always be the case.

If the normal limitations regarding the number of layouts which may be held in two consecutive 96 character buffers are observed then it is necessary, when printing the content of the tape off-line, to switch to single matrix working. This effectively reduces the tape buffer to the size of a printer buffer and therefore the layout store will be able to hold the number of layouts sent to it.

However, it is possible for a print line to straddle two 96 character blocks on the tape, and the line will be recorded all in one 32 word tape block, or straddling two 32 word tape blocks.

In the latter case a print cycle will be lost during the transfer of the first part of the line and the second part of the line to the print line store. This is due to the fact that only one 32 word buffer is being used and this cannot be filled with the balance of the print line until the first part has been transferred to the print line store. During this delay the print cycle is lost.

As this may occur frequently it is advisable to restrict the number of layouts further and to ensure that a D18 is given with every instruction containing layout digits. This will mean that no more than 4 layouts can appear in 4 consecutive ¼ tape buffers, and double matrix working can be used safely. If a lot of short lines are to be printed, this system will use more tape, since a tape block can only contain a maximum of two lines.

When the tape is being printed out, it is necessary to have a means of stopping at the end of the valid information on the tape. This is done by recording on the tape a dummy print instruction specifying E.C.U.7 but no web movement. When the printer finds this instruction, the machine halts. At the same time, a signal is given to the tape unit to rewind set to finish. If E.C.U. 6 is specified, again without web movement, the printer stops but the tape does not rewind. This is mainly of use if it is required to change to a different stationery during a run. Pressing the printer start button allows operations to continue, the data accompanying the E.C.U. 6 instruction being deliberately lost.

7.6 Anelex Printer

7.6.1 The Printer

The printer has up to 160 print positions spaced at 10 to the inch. Print is formed by the action of up to 160 individual hammers each striking one of 62 different symbols of type face carried on a continuously rotating print-roll. The hammers operate through one or more layers of paper and carbon.

Only one set of paper tractors and paper feeding mechanism is provided and so only one web of stationery may be used. Tractors are provided on both sides of the paper and also above and below the print position. Up to six copies can be obtained using suitable stationery, and it is also possible to print on 7 thou. card. The width of the paper must be not less than 4 inches and not greater than 19 inches.

The maximum length of a single form is 17 inches, which corresponds to the maximum permissible length of the Vertical format tape. The minimum length of the format tape corresponds to an 8 inch form, so for forms shorter than this, the format tape must correspond to a multiple of the form length. The minimum spacing between lines of print is 1/6 inch, other spacings are multiples of 1/6 inch. Thus, on a 10 inch form, the maximum possible number of lines is 60.

The "print roll" - carrying 62 symbols of type for each of the 160 print positions - revolves at either 1000 r.p.m. or 1,500 r.p.m., according to the setting of a switch. The time required to print information can be calculated using the following rule:

	75 m.sec.	at 1000 r.p.m.	per line of print
or	55 m.sec.	at 1500 r.p.m	
plus	48 m.sec.	per inch of paper moved.	

7.6.2 Controls

"Stop" and "Start"; these are situated on the printer.

"Set up Paper" and "Page Index"; these are also situated on the printer, and cause the paper feed mechanism to operate until the appropriate punching is found on the Vertical Format Tape. These controls only operate when the printer is in the Stop condition, and are used in setting up the stationery. For "Set up Paper", the punching is 3, for "Page Index", it is 4.

7.6.3 Plugboard layouts.

The simplest arrangement is for the information to be printed on the line in the order in which it was sent out by the program. However, a plugboard can be used which allows a further 3 layouts to be used. The information may be rearranged in any way using the plugboard. Separate output characters may be plugged to two (though not more) positions on the line. Plugboards can be removed and replaced very easily and quickly.

7.6.4 Buffer

The buffer has two matrices, each holding 198 characters in the six-bit internal code. The matrices are used alternately, one being filled by program whilst the printer control reads the contents of the other. Switching of the matrices occurs when a signal is given by the program and the printer has completed all its work on the matrix.

7.6.5 Output to the Buffer

Information to be printed is sent to the buffer, using the output instructions 22, 23, 24, 28 and, less commonly, 18. (Section 7.1 deals with these instructions more fully).

When the 'b' address of an output instruction is zero no printing occurs, but the information is still transferred to the buffer. The ten binary digits of the 'b' address are used as follows:

D18	to Switch buffer matrices.
D19-21	for Selective Print-out.
D22-23	to specify the horizontal layout.
D24-27	to specify the paper movement.

When paper movement is called for in a print instruction, printing will occur immediately after the paper has been moved.

Digits 19 - 21

Eight switches, numbered 0 - 7, are mounted on the printer. A line will only be printed if the switch corresponding to the value of the Digits 19 - 21 is set. It is advisable not to use selective print zero at all, and certainly never when off-line printing is going to be used, since certain print layouts resemble rewind instructions to a tape unit when the selective print digits are zero.

Digits D22 - 23

The values these digits can take are 0-3.

The value 0 gives a straight through connection.

i.e. The 1st character in the buffer is the 1st character on the line, and 2nd character in the buffer is the 2nd character on the line etc.

The values 1 - 3 call the three plugboard layouts. 0 may be used without a plugboard in position.

Digits 24 - 27

These digits can take values 0 - 15 and are used to control the paper movement.

0, 1 & 2 give no movement, 1 line spacing (i.e. paper is moved 1/6th inch) and 2 line spacing (i.e. paper is moved 1/3rd inch).

3 - 13 give paper movement governed by the Vertical format tape. This is an 8-hole punched Mylar (or paper) tape on which each sprocket hole represents a permissible position for a line of print. Thus, the distance between sprocket holes is such that a movement along the Mylar Tape from one sprocket hole to the next represents a movement of the stationery of 1/6th inch. The information on the Mylar tape is duplicated, since only 4 holes are needed to specifiy the numbers 3 to 13. When the paper movement digits are in the range 3 - 13, the stationery and Mylar tape move in synchronism until a punched value on the Mylar tape is equal to the value of the paper movement digits.

14 – 15 Are for off-line working only, and give "stop and rewind the tape" and "stop without rewind" respectively.

7.6.6 Layout Characters

When an instruction addressing the printer contains a non-zero 'b' address, three extra characters are generated and stored in the buffer, following the information characters. These extra characters are: first, a special marker character 100100 (value 36), and secondly, two layout characters containing the 'b' address. These two layout characters are generated by adding two zero digits to the most significant end of the 'b' address and splitting the result into two characters of 6 bits each. When the printer recognises a marker character, the next two characters give the required manner of printing the information preceding the marker character. Any paper movement called for by these two layout characters will take place before the line is printed.

7.6.7 Arrangement of lines in the buffer

Normally a D18 is given with each layout, thus placing lines of print in alternate matrices. However, it is possible to store any number of lines of print in a matrix, provided the total number of information characters, plus layout characters (3 for every line of print), does not exceed 198. Printing of the first line will not take place until the matrices have been switched.

It is not permissible to store a line of print partly in one matrix and partly in the other.

7.6.8 Off-line working

Information may be written onto a tape for subsequent off-line printing.

The output instructions in the program are the same as those for an on-line printer with the following exceptions:

1. The instructions will address a channel which is connected to a tape unit, and not a printer.

2. The Tape unit must be set to write at the beginning of the program. At the end of the program a dummy line with paper feed 14 must be recorded and then the tape must be rewound. This makes the tape rewind to finish at the end of the off-line run.

3. To allow for stationery changes during off-line printing, dummy lines with paper feed 15 may be recorded on the magnetic tape. When these dummy paper feeds are read during the off-line run, the tape and the printer are stopped. This allows stationery, plugboards and selective print switches to be changed as required, before continuing the off-line run by pressing the start button on the tape unit.

7.6.9 Run out of Paper facilities

Two types of "paper out" are available.

1. When the end of the paper has been taken above the lower pair of tractors, the current line will be printed and then the printer will stop, showing failure of the "Ready" light. This "immediate paper out" action is inhibited when the set-up-paper control is pressed, since this control is used before paper is inserted.

2. This second facility is an "Advanced Paper-Out". This facility is initiated when the end of the paper is about 22 inches below the print position. The action is to stop the print out by causing the "Ready" to fail only when a punching of 4 is sensed on the Vertical Format Tape. Thus if Paper Feed 4 is always used as the last line of a form, the action would be to stop the printer after printing the last line of a form. This means that the set up paper and page index controls can then be used when inserting more stationery. This facility can only be cleared by pressing the stop button, and then, to continue printing, the start button must be pressed.

7.6.10 Examples of Print Instructions

A 4 - line name and address is held in registers 300 - 308 in the following manner:-

Register	300	A	.	B	.	J	O
	301	N	E	S	n/c	n/c	n/c
	302	1	3	9	n/c	W	I
	303	N	C	H	E	S	T
	304	E	R	n/c	G	A	R
	305	D	E	N	S	n/c	n/c

Register	306	H A R R O W
	307	M I D D L E
	308	S E X . n/c n/c

The name and address is to be printed at the top of the form, with double spacing between the lines. The first line on the form, which is the first line of the name and address, is equivalent to the line on the Vertical Format Tape punched with value 9.

The sequence of instructions is:

1.	24	300	579	12	9	Name
2.	24	302	0	12	9	Road
3.	24	304	133	12	9	
4.	24	305	130	6	9	Town
5.	24	307	133	12	9	County

Instruction 1 prints the Name on the first line of the form using plugboard layout 0 (i.e. straight-through layout) with a selective print value of 1. Instructions 2 and 3 print the Road two lines below the name, using plugboard layout 0 and selective print 2. Instruction 4 puts the town into the buffer followed by the three special characters (36 char. + 2 layout characters). Instruction 5 puts the County into the buffer, followed by its 3 special characters, and also switches the buffers. Thus, the Town is printed through plugboard layout 0 with selective print 1 two lines below the previous line, and the County is printed two lines lower through plugboard layout 0 and with selective print 2.

All the above 4 lines will be printed <u>only</u> if both the selective print switches 1 and 2 are set. If switch 1 is set, but 2 is not, the name and, two lines below it, the town will be printed; but the road and county will not be printed. If the instructions 1 - 5 were on an off-line tape, then, with both selective print switches 1 and 2 set, the off-line print-out would contain 4 line names and addresses. On another off-line printing run, with only selective print switch 1 set, Names and Towns only would be printed. Thus, for the price of print instructions for a complete form and the setting of selective print switches, the complete form and a summary form (say) can be both be produced off line.

7.6.11 Character repertory

Code	Character	Code	Character
0	0	32	n/c
1	1	33	½
2	2	34	- (minus)
3	3	35	Space
4	4	36	Buffer Control
5	5	37	=
6	6	38	A
7	7	39	B
8	8	40	C
9	9	41	D
10	10	42	E
11	11	43	F
12	12	44	G
13	%	45	H
14	£	46	I
15	&	47	J
16	*	48	K
17	/	49	L
18	. (Stop)	50	M
19	¼	51	N
20	¾	52	O
21	' (Apos)	53	P
22	(	54	Q
23	)	55	R
24	+	56	S
25	X	57	T
26	1/3	58	U
27	$	59	V
28	, (Comma)	60	W
29	_ (Underline)	61	X
30	@	62	Y
31	?	63	Z

7.6.12 <u>Notes</u>

1. The "Set-up-Paper" and "Page Index" controls require that 3 and 4 are both punched on the vertical format tape, if these controls are to be used in the initial positioning of the stationery. These punchings (of 3 and 4) may be used by program, and are not specifically reserved for use solely by the "Set up Paper" and "Page Index" controls.

2. When using function 18 (for example, to put out a name and address) the data may contain sets of 3 characters to act as layout characters. These layouts will be detected and acted upon by the printer in exactly the same way as if they had resulted from a non-zero B address of an instruction addressing the printer.

3. If a matrix is overfilled (i.e. the end of a matrix is reached without layout characters being sent), the data will be printed through layout 0, and preceded by a single line paper feed. Printing will take place whatever the selective print-out setting.

4. To guard against excessive paper feed, due to a paper feed code - which is not on the Vertical Format Tape - being called for by the program, a facility is available which stops the paper after a continuous feed of about 2 seconds. It then prints the line and stops the printer by making the "Ready" fail. The printer can only be started be pressing the stop button before the start button.

5. When the "Advanced Paper out" facility is being used on a form where the last line of print is not always in the same position, (e.g. the position of the total line on an invoice may be related to the number of items on the invoice) it is not possible to give this line a line feed 4. But, so that the facility can be used, it is quite in order to use a dummy print instruction, using a line feed 4 immediately following the total line.

7.7. <u>Output Card Punch</u>

7.7.1. <u>Mechanism</u> (Diagram 1)

I.C.T. (Hollerith) Type 234 gang punch

80-column cards.

Speed : 100 cards per minute

Provision for reading back after punching

Provision for a plugboard

7.7.2. <u>Controls provided on the punch</u>

Start

Halt

Reset : clears punch control

Card run-out : see para 7.

Clear error : see para 8.

7.7.2.1. <u>Indicators provided</u>

Mains on

Start

Halt

Error No data Receiver full	If punch is started but not punching, these indicate three possible reasons.
No pre-punch card No brush card	see below (para. 3)

Test: indicates that some part of the punch control or buffer is on test.

7.7.3. <u>Stations</u> (Diagram 2)

During operation there are three cards within the punch, occupying three 'stations' called pre-punch, brush, and check; at each cycle the cards move to the next station, a fresh card being fed from the hopper to the pre-punch station and the card from the check station being passed to the receiver. Cards are punched while moving from the pre-punch station, to the brush station, and sensed for checking while moving from the brush to the check station.

The No Pre-punch Card and No Brush Card lights indicate that these stations are empty. This is the normal situation before starting; during operation they show a mis-feed or wreck within the punch.

Some notes on card codes, together with the normal EMIDEC code, will be found under Punched Card Reader.

7.7.5. Buffer, Decoder, and Punch Store (Diagram 3)

1. The buffer has two matrices, used alternately. Each holds up to 99 characters in the 6-bit internal code.

2. Information is transferred from the computer to the buffer by means of output instructions (see section 7.1 - Input and Output Instructions). Characters are assembled in the buffer from the 'left-hand' end.

3. A D18 switching signal is given in the last instruction for a batch of data to go into one matrix. This causes three characters (with 6-bit code values 36, 0, 1) to be stored in the buffer following the data characters, and the remaining positions in the matrix to be filled with 'no characters'.

4. The D18 signal also causes the computer output to be switched to the other matrix as soon as it is available.

5. Provided that the punch has been started and no hold-up has occurred, it begins to read out the contents of the matrix as soon as the switch has taken place.

6. Each 6-bit character as it is read from the buffer is converted into the equivalent 12-row card code and passed to a punch store having 80 positions. When the '36' marker character is detected, no further information is read from the buffer for this card, and the contents of the punch store are punched via the plugboard (see para. 7.7.6) at the next punching cycle.

7. It is possible to store information in one matrix to be punched on two (or more) successive cards, provided that the capacity of the matrix is not exceeded. The end of the information for the first card (or any other intermediate card) is marked by a D19 in the instruction. This causes the three characters 36, 0, 2 to be stored in the buffer, but no switch takes place. The last instruction of the last card contains a D18, which has the usual effect. (see 3 above).

8. On reading out from the buffer to the punch store, each '36' marker signals that the preceding information is to be punched into a card. However, only if the two following characters have a 1 will the matrix be switched.

9. If there are more than 80 data characters in the buffer before the '36' marker then the first 80 will be punched and the remainder lost. If the matrix is completely filled with data characters, and a 36 does not appear, then the first 80 characters will be punched, the remainder lost, and the contents of the next matrix may well be lost also.

7.7.6. Plugboard

There is a position on the punch for a plugboard, which allows the arrangement of characters in the punch store to be altered before punching. However, only the one arrangement is available at a time, and duplication of fields by plugging is not permitted. For many purposes a 'straight' plugging is satisfactory, and this is normally used.

The punch will not operate without a plugboard in position. This is valuable when a spare mechanism is available, since a change can be effected simply by taking the plugboard from one and inserting it in the other.

7.7.7. Operation

1. To start, load blank cards into the hopper, press 'Reset', press 'Start'. A card is fed into the pre-punch station and the No Pre-punch Card light goes out.

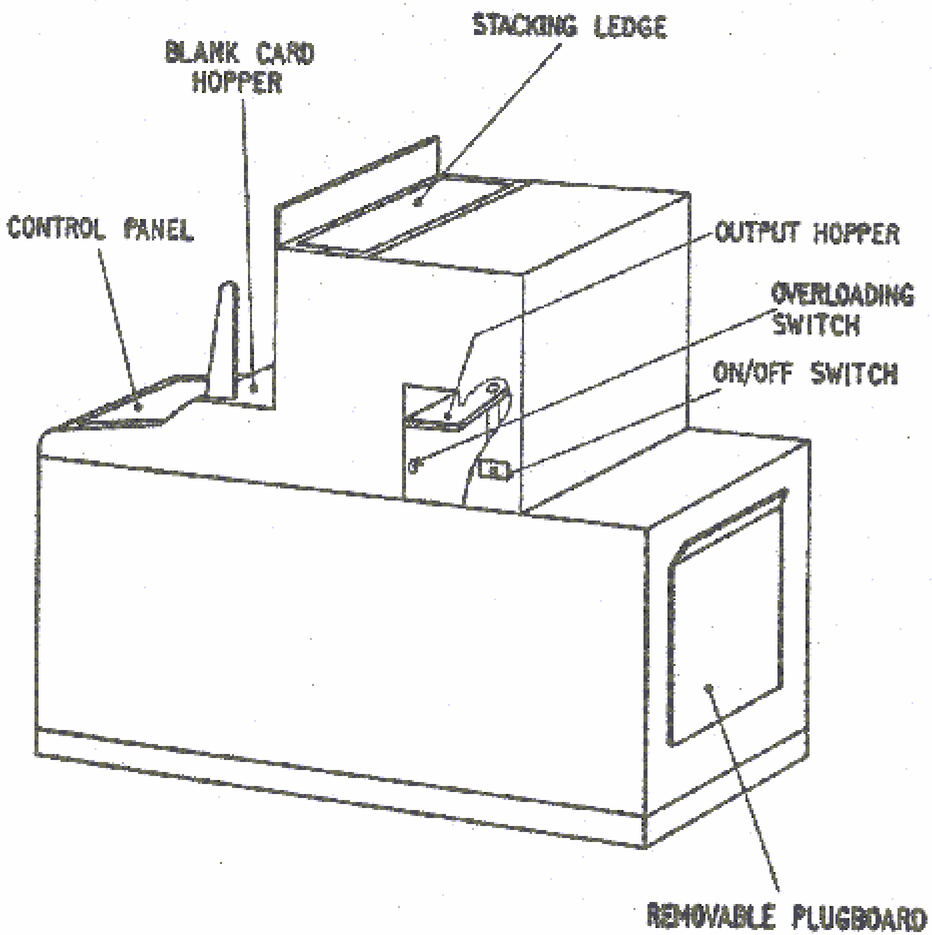
2. When punching is finished, remove any blank cards from the hopper and press Card Run Out. The last three cards are passed to the receiver (including the blank card from the pre-punch station). Remove punched cards from the receiver.

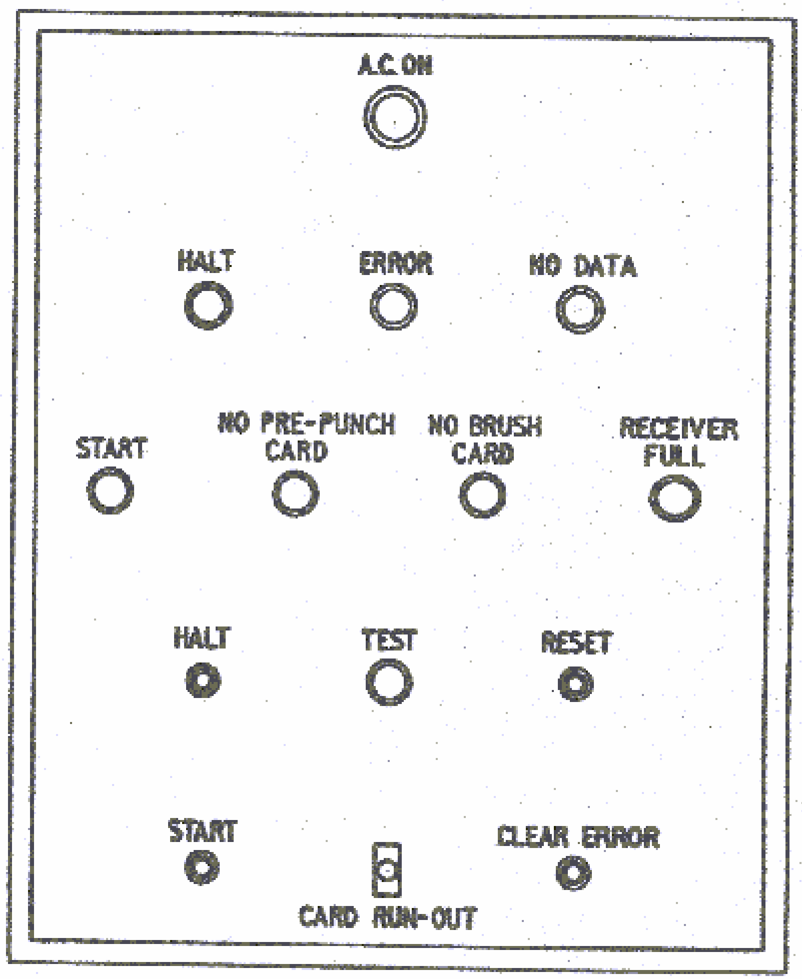
7.7.8. Checking facility

As noted in para. 3 above, a card, after being punched, passes to a sensing station where the punching can be read. There is a check here that the total number of holes punched is correct. If the check fails, the punch halts and displays an Error indication.

In case of error, the procedure is to insert a marker card in the receiver and press the Clear Error button, which allows operation to continue. The suspect card is now immediately following the inserted one.

TL.1063/7.7



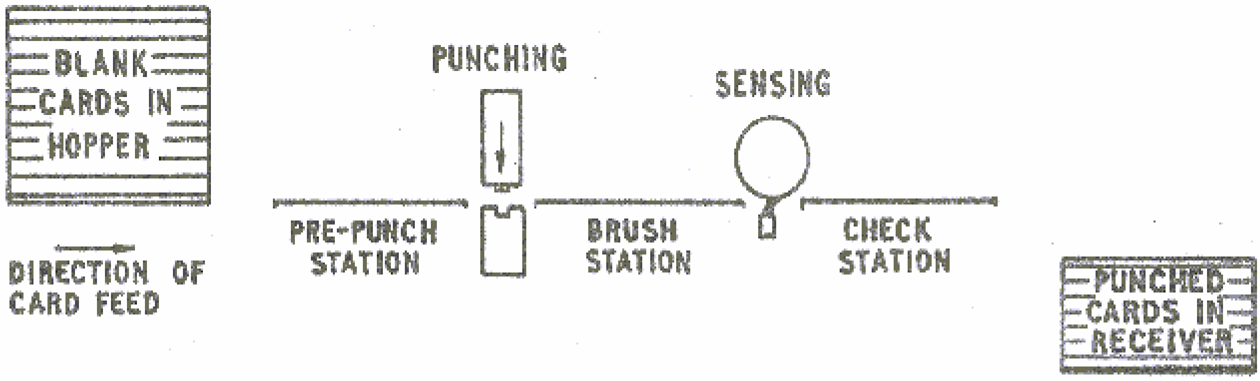


CONTROL PANEL

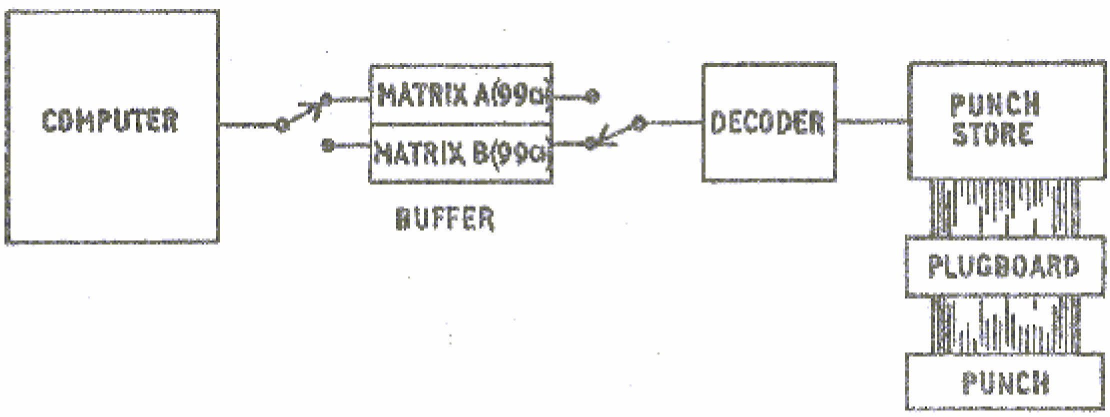
DIAGRAM I

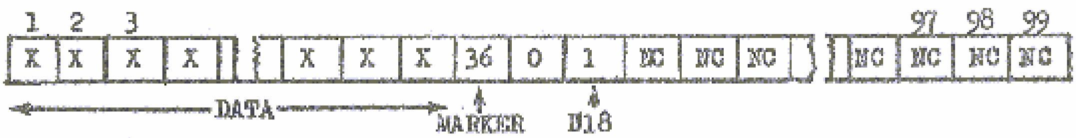
OUTPUT CARD PUNCH

OPERATION OF PUNCH DIAGRAM 2



BUFFER, DECODER AND PUNCH STORE DIAGRAM 3





Normal arrangement of characters in matrix.

Arrangement of characters for two cards in one matrix.

7.8. Paper Tape Punch (300 characters per second)

7.8.1. Creed Model 3000 Punch (Diagram 1)

Channels: 5, 6, 7 or 8 hole codes.
Speed: 300 characters per second
Provision for read - back check after punching.

7.8.2. Controls provided on the punch

Start:	initiates punching
Stop:	halts the punching process after the current character.
Run-Out:	blank tape is punched while the button is held down (See 7.1)
Pass fault:	continue operation after detection of an error (See para. 5)
Reset:	clears buffer.

7.8.2.1. Indicators provided

Mains on
Tape empty (see para. 6)

Punch fault:	read-back after punching check indicates an error.
Punch failed:	in case of error, a set of lights indicates the channel (or channels) in which a hole was not punched.
False punch:	in case of error, a set of lights indicates the channel (or channels) in which a spurious hole appears.

7.8.3. Paper tape codes

See para. 3 of 'Paper Tape Reader' for some notes on codes and details of a sample 5-hole code.

7.8.4. Buffer and Decoder (Diagram 2)

1. The buffer has two matrices each holding up to 99 characters in the 6-bit internal code.

2. Information is transferred from the computer into the buffer by means of output instructions (see section - Input and Output Instructions). Characters are assembled in the buffer from the 'left-hand' end.

3. The information in the buffer will include control characters (e.g. Carriage Return/Line Feed) if the paper tape is later to be transcribed on a teleprinter or electric typewriter. (See 7.2).

4. A D18 switching signal is given in the last output instruction of the batch of data which is to go into one matrix. This causes three characters (having 6-bit code values 36, 0 and 1) to be stored in the buffer following the last data characters, and the remaining positions in the matrix to be filled with 'no characters'.

5. The D18 signal also causes the computer output to be switched to the other matrix as soon as its previous contents have been punched. (See 7.3).

6. Provided that the Start button has been pressed, the punch begins to read out the contents of the matrix as soon as the switch of matrices is complete.

7. Each 6-bit character from the buffer is passed to a decoder, which converts it from the internal code to the paper tape code being used. In the case of a 5-hole code any case-shift characters required are inserted at this point; one character from the buffer may therefore result in two punchings on the tape. (See 7.4).

8. When the '36' character that marks the end of the information in the matrix is detected, the remainder of the matrix is cleared.

9. When the end of the matrix is reached, the switch of matrices takes place as soon as the computer gives the D18 signal.

7.8.5. <u>Read-back check on punching</u>

A photo-electric reading station is provided which allows each punching to be read back and compared with the original information. (There is a gap of three character positions between punching and reading).

When an error is detected the punch stops with the Punch Fault lamp lighted, and in the sets of indicator lamps the details of the fault (holes missing or in excess - see para. 2). The procedure is then as follows:

1. Note details of fault (holes missing or in excess)
2. Press Stop button
3. Press Pass Fault button
4. Press Run Out button until blank tape is visible.

The faulty character is now the fifth before the blank tape (or sixth if the penultimate punching is a case shift in 5-hole code), and may be marked for later attention.

7.8.6. Tape Empty

When the end of the supply reel of tape is approaching, the punch stops with the Tape Empty lamp lit. In this state punching will continue so long as the Start button is pressed and held down.

A conditional halt should be provided in the program at each point where the output punching can conveniently be terminated, and when the Tape Empty state is reached the condition switch is set and the Start button on the punch pressed until all punching up to the conditional halt is complete. A new reel of tape can then be loaded.

7.8.7. Notes

1. With a 5-hole code, the 'blank tape' combination is often used as figure or letter shift. In such a case the Run-Out button also causes the decoder to be set in the appropriate state.

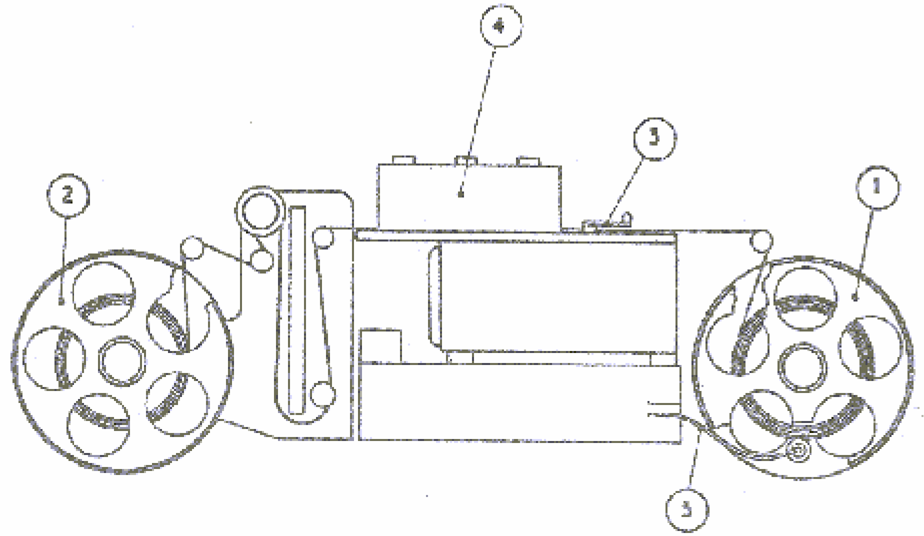
2. If the punched tape is to be interpreted on a Creed Model 75 Teleprinter, the time taken for a carriage return is in certain circumstances longer than the time for reading one punching. To avoid the difficulties which can arise from this it is advisable to punch a further character after every Carriage Return control character; if the line feed is combined with the carriage return a dummy punching such as Figures or Letters shift should be used.

3. The D18 signal should be given when there are not more than 95 characters in the matrix (including any characters put out by the instruction in which the D18 occurs).

4. Any 6-bit character in the buffer which has no equivalent in the paper tape code will give no punching on the tape.

5. If appropriate E.C.U. and paper movement digits are also given with each D18, the information can be printed on a Samastronic (for example in program testing) by plugging the printer to the punch channel. The extra digits do not affect the working of the punch.

6. The conventional channel numbers for paper tape punches are 14 and 15.



1. LOADING SPOOL
2. TAKE-UP SPOOL
3. TAPE THREADING MECHANISM
4. PUNCH BLOCK
5 TAPE EMPTY SENSING MECHANISM

CONTROL PANEL

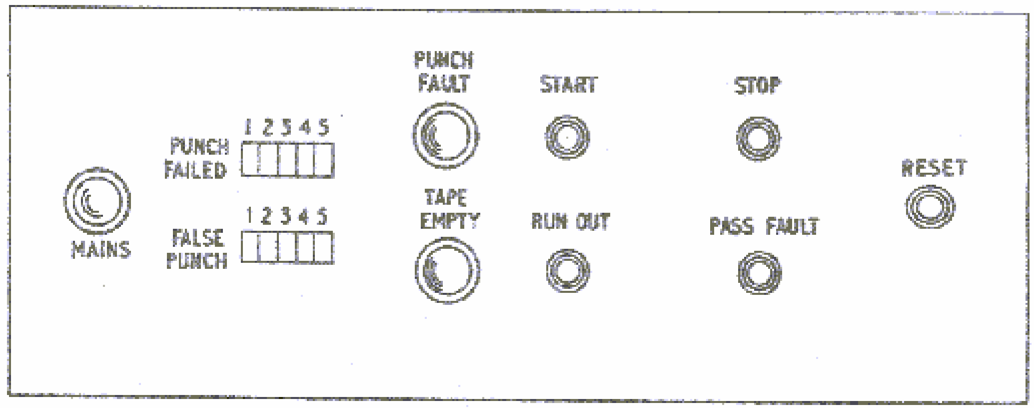
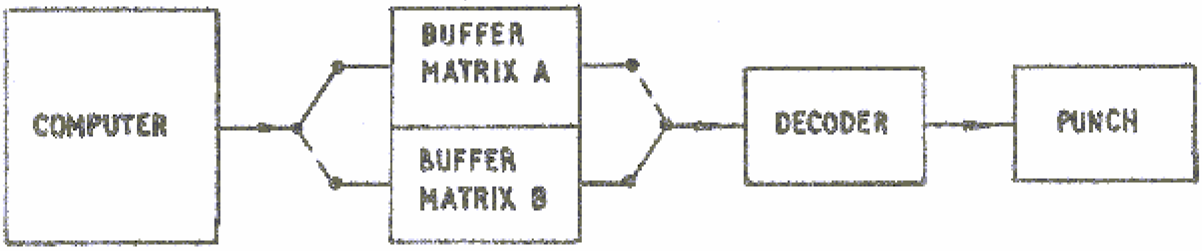


DIAGRAM I CREED 3000 PAPER TAPE PUNCH

BUFFER & DECODER DIAGRAM 2



TL.1063/7.B

7.9 **Paper Tape Punch** (25 characters per second)

7.9.1. **Creed Model 25 Reperforator** (Diagram 1)

Channels: 5, 6 or 7 hole tape.

Speed: 25 characters per second.

7.9.2. **Controls provided on the punch**

Mains switch

Run-out : blank tape is punched while the button is held down (see 5.1)

Stop : halts the process of punching

Start : initiates punching

7.9.3. **Paper tape codes**

See para. 3 of 'Paper Tape Reader'

7.9.4. **Buffer and Decoder** (Diagram 2)

1. The buffer has one matrix holding up to 99 characters in the 6-bit internal code.

2. Information is transferred from the computer into the buffer by means of output instructions (see section - Input and Output Instructions). Characters are assembled in the buffer from the 'left-hand' end.

3. The information in the buffer will include control characters (e.g. Carriage Return/Line Feed), if the paper tape is later to be transcribed on a teleprinter or electric typewriter. (See 5.2).

4. A D18 switching signal is given in the last output instruction of the batch of data which is to go into the buffer. This causes three characters (having 6-bit code values 36, 0 and 1) to be stored in the buffer following the last data characters, and the remaining positions to be filled with 'no characters'.

5. The D18 signal also causes the buffer to be switched from the computer to the punch. While the contents are being punched the buffer is not accessible to the computer.

6. Provided that the Start button has been pressed, the punch begins to read out the contents of the buffer as soon as the switch has taken place.

7. Each 6-bit character from the buffer is passed to a decoder, which converts it from the internal code to the paper tape code being used. In the case of a 5-hole code any case-shift characters required are inserted at this point; one character from the buffer may therefore result in two punchings on the tape (See 5.3).

8. When the '36' character that marks the end of the information in the buffer is detected, the two following characters are read out but not punched. The 'no characters' which fill the remainder of the buffer do not result in any punching.

9. When the last character has been read, the buffer is switched back to receive further output from the computer.

7.9.5. **Notes**

1. When, in a 5-hole code, the 'blank tape' combination is used as figure or letter shift (as is usually the case), the run-out button also causes the decoder to be set in the appropriate state.

2. If the tape is to be interpreted on a Creed Model 75 Teleprinter, the time taken for a carriage return is in certain circumstances more than the reading time for one punching. Therefore unless separate Carriage Return and Line Feed controls are included in the code a dummy character should be punched after each carriage return; figure or letter shift are suitable.

3. Any 6-bit character in the buffer which has no equivalent in the paper tape code will not cause any punching. Such characters are dealt with rapidly.

4. The punch can prepare two duplicate tapes simultaneously if desired.

5. The conventional channel numbers for paper tape punches are 14 and 15.

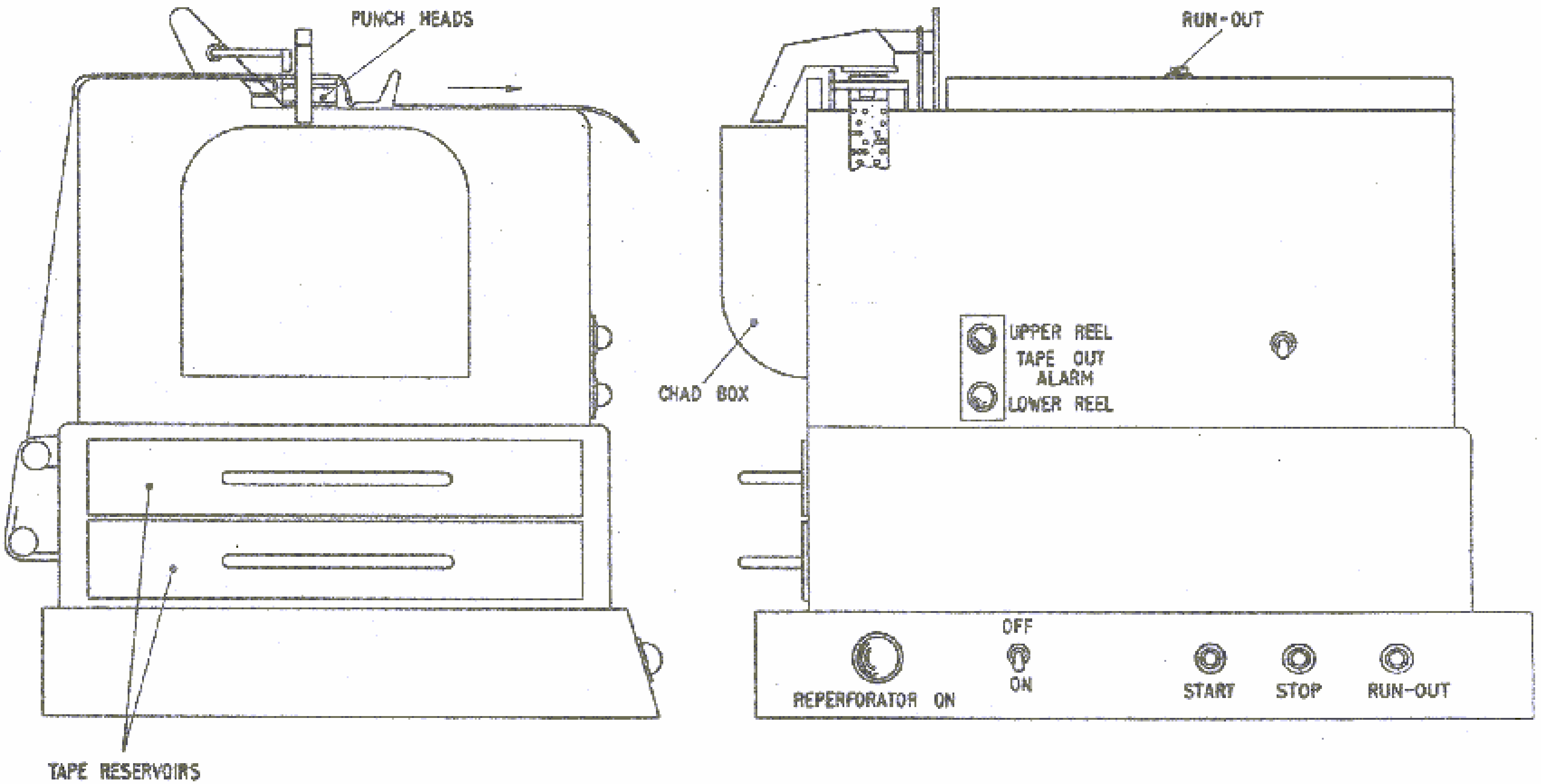
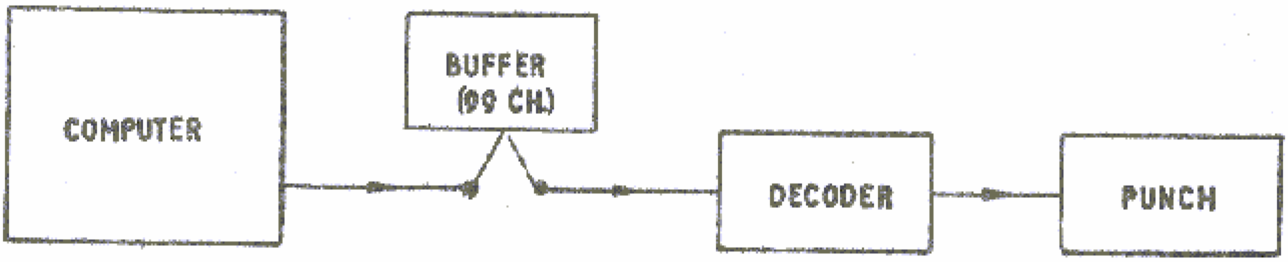


DIAGRAM I **CREED 25 PAPER TAPE REPERFORATOR**

BUFFER & DECODER DIAGRAM 2

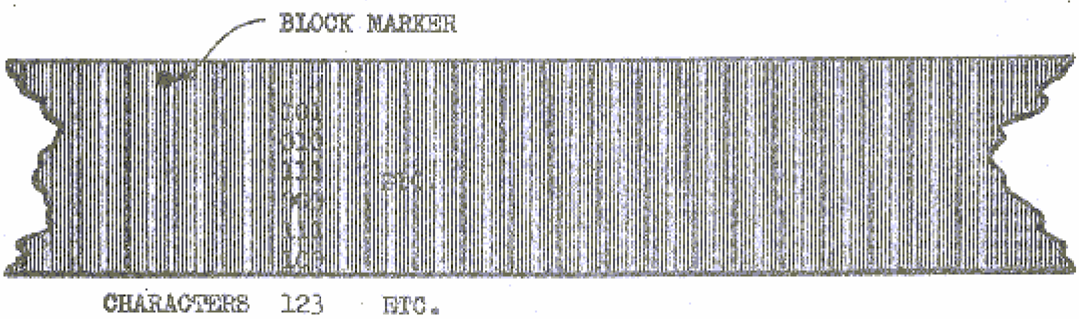


7.10. 40 Kc. MAGNETIC TAPES

7.10.1. Specification and Method of Recording Information

Running speed of tape 120 in. per second } nominal
Net character transfer rate } 13.5 Kc/s } nominal

Information is recorded in blocks of 198 characters, each of 6 bits; of these, normally, 192 are information characters and 6 are for checking purposes. The positions of the blocks are pre-determined, the block markers being recorded before a tape is issued for use. The characters are recorded serially, across the tape, thus:-



The six information tracks are duplicated. There are, in fact, 16 tracks across the tape, allocated as follows:-

6 information tracks (duplicated)
3 clock tracks
1 block marker track.

The clock tracks are used for timing purposes.

On reading a tape, the 12 information tracks are read simultaneously, and the signals from each pair of tracks are mixed to minimise the chance of a bit being corrupted; if one track fails, the signal from the other is sufficient to give correct information. Provision is included for back spacing and re-reading under program control.

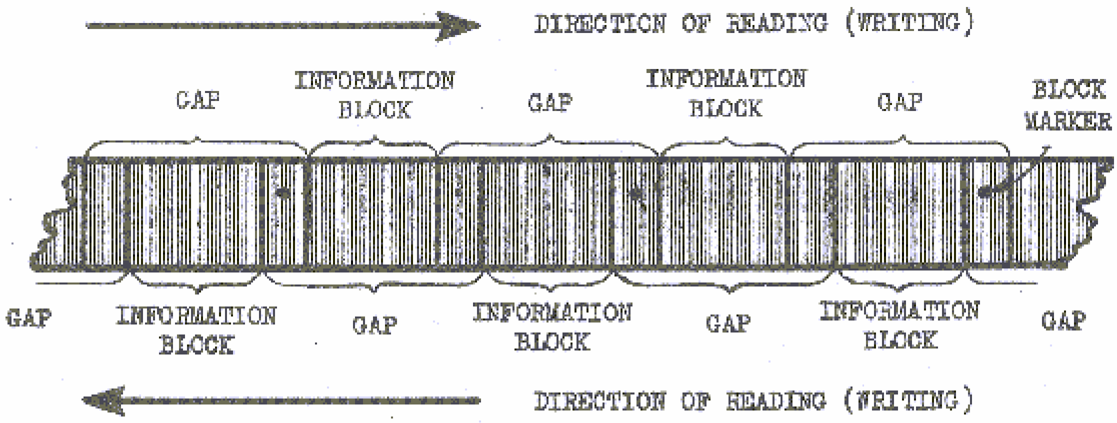
Blocks are recorded in both directions on the tape. The tape runs forward for its whole length, recording or reading on alternate blocks. When it reaches the end, it is automatically reversed, and recording or reading now takes place on the blocks in between; alternatively, it may be reversed by the program before the end is reached.

This is done:

(a) to allow space for stop/starting at any point on the tape

(b) to minimise rewinding time (a tape must always be rewound to its original position after use).

The process may be shown as follows:-



Each information block occupies approximately 0.6 inches, and the distance between the starting points of consecutive blocks read in the same direction is 1.76 inches (i.e. the pattern is repeated each 1.76 inches). The gaps shown between information blocks are the positions on which the braking rollers act.

Tapes are normally available in five standard sizes. These are shown below, together with the approximate running times and guaranteed minimum capacities:-

Length	300'	600'	1200'	2400'	3600'
Capacity:					
16 word blocks	5,700	13,500	29,100	60,300	91,500
32 word blocks	2,850	6,750	14,550	30,150	45,750
Running time:					
(go and return)	1 min.	2 min.	4 min.	8 min.	12 min.

Certain other sizes can be made available by special request.

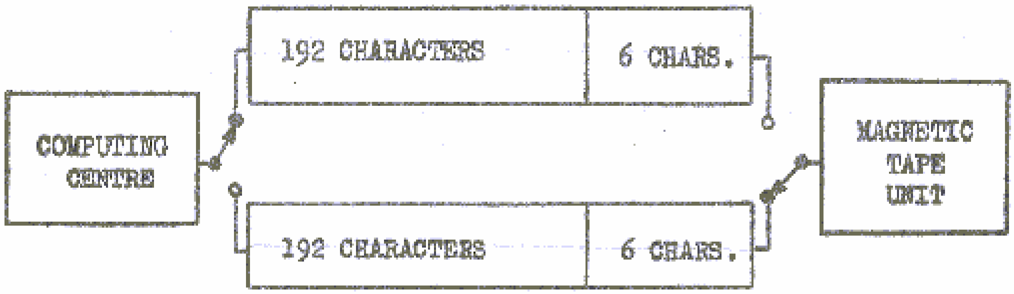
To maintain full speed on a tape unit, the computer must process the information for, or from, a tape block of 192 characters within 14.7 msecs average. If the computer exceeds this, a signal will be given to stop the tape.

It is not possible to read from and write (record) onto a tape during the same run. When a tape is loaded and the unit is started, the tape is automatically only available to be read from. If writing (recording) is to take place, a special control signal must be sent by program which will set the unit into the 'write' condition. If, subsequently, the tape is to be read back before being removed, another signal returning the unit to the 'read' condition is needed. Both these control signals automatically rewind the tape to the beginning (by the shortest route, i.e. if the tape is moving in the forward direction, it is simply reversed without first running through to the end). A third control signal is used when the unit has been finished with, to rewind and set to 'finish' condition. Rewind, set to 'read' or 'write' operate the tape at normal running speed. Rewind set to 'finish' moves the tape half as fast again.

7.10.2 The Buffer System

The buffer for the increased data rate magnetic tape unit contains two matrices of 198 characters. It may be used in exactly the same way as the earlier magnetic tape buffer, that is, as four $\frac{1}{4}$-buffers of 99 characters (see section 7.2.2). Programs written for the earlier system will function on the new one, except that parity failures can no longer occur.

However, to realize the full benefit of the increased speed, it is desirable that the program should work in terms of the blocks used on the tape itself, i.e. 198 characters. A block will conveniently be divided into 192 characters (32 words) of data, and 6 characters (1 word) of check information. In this arrangement, the buffer will appear as shown overleaf.



7.10.3 Reading and Writing on Tape.

If the tape is being used in the same manner as the earlier system, then the same instructions will be used for reading and writing (see 7.2.3.).

However, subroutines are available which will compile the check information on writing, and record the block on tape, and, on input, will read the block and test the check information. In the event of a check failure, the subroutine will back-space the tape and re-read, and, if the error is still not cleared by this means, will report it in a standard manner.

Since the process of back-spacing may in some cases be awkward, it is recommended that use be made of the subroutines wherever possible. Detailed directions for their use are available separately.

7.10.4. Control Signals

The control facilities provided allow the program to

(a) set a deck into the forward or reverse direction.

(b) rewind the tape and set the deck into the 'finish', 'read' or 'write' condition.

All these controls are exercised by means of digits in the B address of an input/output instruction addressing the appropriate channel.

The direction change, to reverse or forward, is signalled by a D22 or D23, together with a D18. Thus:-

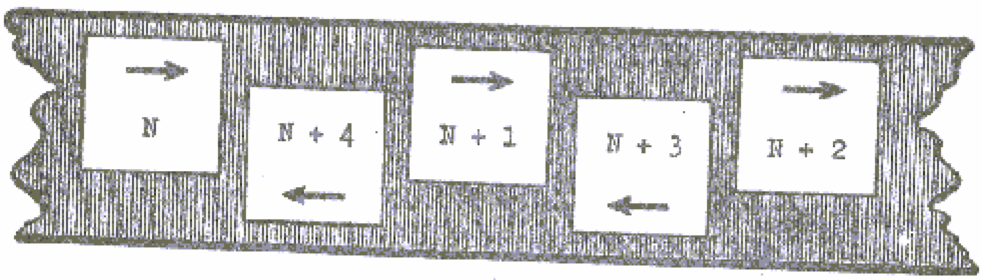
1. Set M.T.U. 1 to 'reverse'
   19.0.17.1.1.
2. Set M.T.U.2 to 'forward'
   19.0.33.1.2.

If the unit is already in the state to which the signal would set it, the instruction will have no effect. If a change is entailed, the action is as follows:-

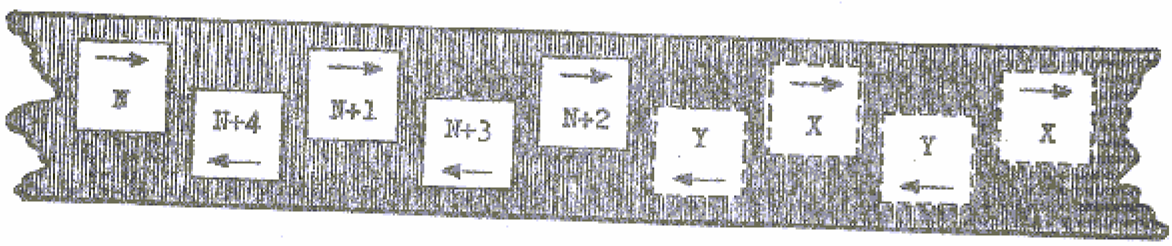
(1) On a writing deck, any complete 198-character tape blocks in the buffer are written onto the tape, and the change takes place before further data is accepted by the buffer.

(2) On a reading deck, the buffer is filled and cleared, the change takes place, and the buffer is filled and cleared again before further data may be read from it.

In this way, the changes during writing and reading are given the same effect.



Effect of giving a 'change to reverse' signal after block N+2 on writing.



Effect of giving a 'change to reverse' signal after block N+2 on reading. Blocks X, X and Y, Y are read into the buffer and cleared; block N+3 is the next block available to the computer.

Changes of direction within a few blocks of the beginning of the tape and the automatic turn-around point cause complications, and should be avoided. The read/write subroutines include provision for reversing the tape when required and avoiding these complications. However, they do not include any provision for rewinds.

To rewind a tape to the beginning, and set it to 'finish', 'read' or 'write', a bit is placed in the D25, 26 or 27 position of the instruction concerned, together with a D18.

Thus:-

1. Rewind the tape on M.T.U. 1 and set it to finish

   19 . 0. 129. 1. 1.

2. Rewind M.T.U. 2, set to read

   19. 0. 257. 1. 2

3. Rewind M.T.U. 2 set to write

   19. 0. 513. 1. 2

It will be seen that the function used is input decimal to register 0 in all cases. In practice, any input/output function can be used, but, unless care is taken, other functions can overwrite wanted data, either in the store or on the tape. This cannot happen using an instruction as suggested above and, therefore, function 19 is normally used. It must be added that, for control purposes only, it is in order to address a deck, which is in the output ('write') state, with an input function.

Certain rules associated with control signals must be obeyed. (Rules 4 and 5 do not apply when the read/write subroutines are in use).

The rules are:-

1. After a rewind, set to write, the tape must be written on (i.e. at least one tape block must be recorded) before a further control signal is given to the same deck.
2. After a rewind, set to finish, the tape unit cannot again be addressed by the computer until it has been manually restarted.
3. Only one control signal may be given in one instruction. Digits D19-21 must be zero.

4. After a rewind or change of direction, there must be at least one instruction (of any sort) in the program before the next instruction addressing the deck just addressed. This is because the electronics concerned with the buffer take a little time to absorb the control instruction; if the very next instruction tried to address the same deck, the buffer etc. would not be quite ready and the instruction might not be carried out correctly.

5. Before giving a rewind signal to a deck on which writing has been taking place, it is necessary to record one extra (dummy) 16 word block. This is because, when the rewind signal is given, it is carried out as soon as recording from the ½-buffer connected to the tape unit at the time is completed. Should the rewind signal be associated with the second of a pair of ½ buffers, the information in the previous half buffer will be successfully recorded, but the information in the first half of the buffer connected to the computer at the time the control signal is given will be lost. It would be possible to avoid losing information in this way by ensuring that any rewind signal coincides with an odd numbered D18. This, however, means counting blocks, and it is easier always to output the one extra block; if a block is then lost, it will be the dummy block and not the last data block.

7.10.5. <u>Checks on Information</u>

It is normally essential to ensure that the tape writing and reading processes have been carried out correctly. To this end, check information is included in the data recorded on the tape, and is tested on reading. In 7.2.5 is explained how this is done with the earlier tape system, and these principles should be followed if the read/write subroutines are not in use. (However, parity is no longer available as a means of checking).

The read/write subroutines provide check information with each 32-word block recorded on tape, consisting of check total, block number, and run number. This is recorded in the remaining 6 characters and, on reading, it is checked. If any discrepancy is discovered, a further attempt is made at reading; if this is still incorrect, a report is printed, and a signal is given to the program.

7.10.6. Other input/output Functions

It is possible to use normal input/output functions 19 - 24 and 28 with tapes. The common application is in preparing tapes for off-line printing, (see 7.5.6 and 7.6.8.).

If a tape prepared by this method is to be read back into the computer, the equivalent input functions will covert the data back into binary form.