6. TABLES

6.1. Purpose
The purpose of tables is to specify:

i. for the unpack actions (50 and 51) the 'destination' locations and changes in format required.

ii. for the edit and condense actions (52 and 53) the 'source' locations and changes in format required.

6.2. Table entry sheets - general
Tables are entered on table entry sheets (see specimens, Appendices C & D). The entries in the heading block on this sheet are described in section 6.3.

In the main body of the sheet one table entry (normally the details for one word of information) is entered on each line. These entries are given a pre-printed serial number, so that any entry may be referred to uniquely by quoting its section number and serial number within the section.

The entries in the various columns are described in 6.4.

If a table is too long to go on one sheet, continuation sheets may be used (see Appendices C & D for specimens). The serial numbers must be arranged to follow in sequence right through the table, but no serial number may exceed 96.

6.3. Heading block entries
6.3.1. Section number
This indicates the number of the section which the table occupies. It must be in the range 100 - 998 (see section 4.3.2).
6.3.2. **Type**
This indicates the action using the table:
1 = EDIT (action 52)
2 = Condense (action 53)
3 = Unpack fixed field data (action 50)
4 = Unpack variable field data (action 51)
(For tables used by more than one action see section 6.5.)

6.3.3. **Name and Notes**
These may contain further description of the table and its uses.

6.4. **Table entries – details of columns**

6.4.1. **First Item**
This column holds the location to which the table entry applies, in reference and item form. It may not refer to a transit area. In Condense tables, a sequence of items may be described by one table entry if they are to be selected from consecutive locations and dealt with in the same way. In this case this column holds the source locations of the **first** item.

6.4.2. **Last Item**
If a sequence of items is being described by a single entry in a condense table, this column holds the source location of the **last** item. (The reference must be the same as for the “first item”). If the “first item” is greater than the “last item” no condensing will take place for this table entry.
In all other cases this column should be left blank.

6.4.3. **Short/Long/Alpha**
Each table entry must have this column completed. It indicates the form of the item(s) in the location(s) specified at the beginning of the entry, as follows:

Amendment No. 10
Section 6.4.4.

0 = short numeric form
1 = long numeric form
2 = alpha form

6.4.6. Single / to End
This column may be used in the last entry on type 4 (unpack variable) tables only. If the entry refers to a single item the column is left blank; if the entry refers to a sequence of items terminated by a line end or block end, 1 must be entered in the column.

6.4.5. Number (of characters)
This column may be used in type 1 (edit) and type 3 (unpack fixed) tables only.
When used for unpack fixed, it specifies the number of characters in the item. If no entry is made, 16 characters will be unpacked. When used for edit, it specifies the number of characters of the 'source' item to be copied to the transit area, counting characters from the more significant end of the word. (See section 6.4.6 for an example). If no entry is made 16 characters will be edited if the item is in numeric form and none if it is alpha form.

The maximum number of characters which may be specified is dependent on the short/long/alpha entry as follows:

<table>
<thead>
<tr>
<th>Short/long/alpha</th>
<th>Maximum number of characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
</tr>
</tbody>
</table>

6.4.6. Discarded
This column may be used in type 1 (edit) tables only. It enables selected characters from a word to be edited, by specifying how many characters in the item are to be discarded before copying begins, counting from the more significant end.

Amendment No. 17
June, 1963.
Example:
Location N contains CODES in alpha.
Various possible combinations which can be edited by one table entry are given in the following table:

<table>
<thead>
<tr>
<th>To be edited</th>
<th>Number of characters</th>
<th>Number to be discarded</th>
</tr>
</thead>
<tbody>
<tr>
<td>COD</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>ODS</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>DNS</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>DB</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>S</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

6.4.7. **Spaces**

This column may be used in type 1 (edit) tables only. It enables space characters to be inserted before any item that is being edited. The maximum number of spaces that can be specified in this column in one table entry is 15. (However, by editing a zero compartment numerically, it would be possible to specify 25 spaces with one table entry).

6.4.8. **Zeros**

This column may be used in type 1 (edit) tables only. It enables non-significant zeros to be represented by spaces instead of zeros for the specified number of characters, counting from the more significant end. (The entry in the column must not exceed the entry in the 'number' of characters only).

6.4.9. **Sign**

For type 1 (edit) tables, 0, 1, or 2 may be entered in this column. 0 means item will not be signed in its destination (whether or not the sign bit is set in the source). 1 means item is signed and if it is negative the sign is required in card form (i.e., 1 is added to the control quartet of the most significant character of the field).
2 means item is signed and the sign is required as a separate character (the least significant in the field) which will be a space character if the item is positive and a negative sign if the item is negative.

For type 2 (condensed) tables, 0 or 2 may be entered in this column. 0 means item will not be signed in its destination (whether or not the sign bit is set in the source).

2 means item is signed and if it is negative the sign is required as a separate negative sign character (the least significant in the field).

For type 3 (unpack fixed) tables, 0 or 1 may be entered in this column.

0 means item will not have the sign bit in the destination.

1 means item will have the sign bit in the destination set if the item is to be in numeric form and the most significant character in the source has a control quartet of the form 4p + 1.

For type 4 (unpack variable) tables, this column is left blank. The sign bit in the destination is set if the item is to be in numeric form and any character in the source is a negative sign character.

6.4.10. Sterling

This column may be used in type 1 (edit) tables only. Any sterling amount that is being edited, provided that it is being edited as pounds, shillings and pence, may have this column set at 1 in each of the three entries. If this is done the entry in the "Shillings" column must also be equal to the entry in the "No." column for the shillings item.

The effect of this is as follows:

If the pounds item contains significant digits one non-significant zero will be suppressed for the shillings item, otherwise all non-significant zeroes will be suppressed for the shilling item.
The sterling indicator must be submitted in three consecutive table entries, but if several consecutive tables are obeyed as one the three sterling entries can be split by a new table heading.

6.4.11 Last table entry
The last table entry for any table must have 1 in this column. No other table entry should have an entry in this column.
Several tables may be submitted under the same directive (i.e. on the same sheet) provided that each has 1 in this column for its last entry. Also several tables can be obeyed as one by omitting the last table entry from all but the final table. All tables in a set must be of the same type.

6.5 Use of Tables for more than one action
6.5.1 It is possible under certain circumstances to use the same table for:
(a) condense and unpack variable
(b) edit and unpack fixed
6.5.2 Condense and Unpack Variable
This table has entries as specified in section 6.4. for condense tables except that it may have in addition an entry in the single/B.E. column. The 'table type' in the heading block is 2.
The circumstances in which the same table can be used are:
(i) If each item has a separate table entry (so that the 'Last item' and 'single/to B.E.' columns are not used).
(ii) If the table consists of one entry only, in which the 'last item' and 'single/to B.E.' columns are both used.
(iii) If the table consists of several separate entries as in (i) followed by one entry as in (ii).

Amendment No. 10.
March, 1963
6.5.3. Edit and Unpack Fixed

This table has entries as specified in section 6.4. for edit tables.

The 'table type' in the heading block is 1.

Two points should be noted:
(i) If two items are edited from one location (e.g. using the 'discard characters' facility) the original contents of the location will not be re-formed on unpacking; the location will contain simply the last item to be unpacked.
(ii) If the 'spaces' column is used for edit, these spaces will not be removed on unpacking.

6.6. Note on Edit Tables

The following rule must be observed when writing an edit table:

If \( x \) = the sum of entries in 'number' column for all table entries up to and including the last table entry

\( y \) = sum of entries in 'spaces' column for all table entries up to and including the last table entry.

\( z \) = number of times 'sign' column contains entry '2',

then \( x + y + z \) must be a multiple of 5 to avoid having spurious characters in the last word.

To achieve this, extra space characters may be edited before the line-end or block-end character.

If, however, this character is required in the 121st or 161st character position (see section 13.6), then a word may be edited of which the line-end or block-end is the most significant character.

If zero occurs in the number column and the item is not in alpha form 16 should be added to \( x \).
7. **CONSTANTS**

7.1. **General**

In Intercode the need for holding many constants is avoided by the use of literals. Other constants may however be required (e.g., alphabetic constants and complete tables of numbers) and these are written on constant sheets - see specimen (Appendices C & D).

The entries in the heading block on this sheet are described in section 7.2. In the main body of the sheet one constant is entered on each line. These lines are given pre-printed serial numbers so that any constant may be referred to uniquely by quoting its section number and serial number within the section.

If a constants section is too long to go on one sheet continuation sheets may be used (see Appendices C & D for specimens). The serial numbers must be arranged to follow in sequence right through the section, but no serial number may exceed 96 (long constants) or 97 (short constants).

7.2. **Heading block entries**

7.2.1. **Section number**

This indicates the number of the section which the constants entered below are to occupy. It must be in the range 100 - 998 (see section 4.3.2).

7.2.2. **Type**

This indicates the use of the section as follows:

- 0 means each constant is to go in a short word.
- 1 means each constant is to go in a long word.

In fact there are two different types of sheets as the pre-printed serial numbers differ in the two cases (see specimens).
7.3. Details of method of entering constants

7.3.1. Type of constants (normal)

The following types of constant may be entered:

- Alpha constants
- 'Decimal' constants
- Binary patterns
- Binary numbers
- Relative address constants

Every constant must be of one of these types or of a combination of them and is entered as described in sections 7.3.2. to 7.3.7. below. The first of a group of constants of the same type must be preceded by a directive letter as described below. The first constant in a section must be preceded by a directive letter.

7.3.2. Alpha constants

Each symbol will be represented in the store in alpha form, i.e. as an octet. The six symbols (, "line end"
"Block end") are not permitted; if it is required to hold (for example) a block end as a constant, it must be written in its control and basic quartet form i.e. 774. The first of a group of alpha constants must be preceded by the directive (A).

Example:

```
  0 (A) MR. JO
  2 NMS (B) S
```

7.3.3. 'Decimal' constants

Each symbol will be represented in the store in numeric form, i.e. as a quartet. Negative constants are permitted and will be held in sign and modulus form.

Amendment No.10
The first of a group of 'decimal' constants must be preceded by the directive (D).

Example:

<table>
<thead>
<tr>
<th></th>
<th>(D)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>178</td>
</tr>
<tr>
<td>1</td>
<td>193014-</td>
<td></td>
</tr>
</tbody>
</table>

7.3.4 Binary patterns

Each symbol, 0 or 1, will be represented in the store by a single bit. Negative constants are permitted and will be held in sign and modulus form. The first of a group of binary pattern constants must be preceded by the directive (P).

Example:

<table>
<thead>
<tr>
<th></th>
<th>(P)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td>10100-</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

7.3.5 Binary numbers

These are submitted in decimal, converted, and held in store in binary. Negative constants are permitted and will be held in sign and modulus form. The first of a group of binary number constants must be preceded by the directive (D).

Example:

<table>
<thead>
<tr>
<th></th>
<th>(D)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td></td>
<td>8191</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>87</td>
</tr>
</tbody>
</table>

7.3.6 Relative address constants

These are indicated by the directive (R) or (O) and are submitted in the form reference/colon/item. The item must be positive, and the reference must not be a transit area. Where the reference is a procedure, or table, or constants section, the item measures distance from the procedure start in the computer code expansion, which is one-for-one with the Intercode for tables and constants.

R and O are interchangeable for a constant which is not 'mixed' (see 7.3.7), but in a mixed constant the R directive yields a 15-bit address and the O directive a 16-bit address (essential for computers with more than 4 divisions).
7.3.7 Mixed constants

It is possible to have 'mixed' constants in which several of the above types appear in one word. Each type must be introduced on the first line by its appropriate directive. In subsequent lines the directives are not required provided each type occupies the same part of the word as on the first line; the types are then separated on the sheet by commas. Thus it may be necessary to write non-significant zeros on the first line to indicate the size of the space to be allotted to a particular type.

In the first line of a group of mixed constants, if the directive B occurs, it must be followed (within its brackets) by the number of bits the type is to occupy.

Relative address constants (which occupy 15 bits for an R directive and 16 bits for a C directive) may only occur at the least significant end of a mixed constant.

Negative decimal or binary constants are permitted and held in sign and modulus form if they are the most significant type in the constant; otherwise in the form of complements.

Example:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(D) 0123-</td>
<td>(99)511</td>
</tr>
<tr>
<td>2</td>
<td>7014,</td>
<td>32,</td>
</tr>
</tbody>
</table>
Section 8

8. PROGRAMS SILENTS

8.1 General

As described in section 3, the sheets on which a programme is submitted to the translator are:

Programme Heading Parameters sheet
Section Description sheet
Coding sheet
Table Entry sheet
Constants sheet
Programme End sheet
Start and End of reel sheets

The uses of the coding, table entry, and constants sheets have already been described (see sections 4, 6 and 7). The remaining sheets are described below.

For specimens, see Appendix C.

8.2 The Programme Heading Parameters Sheet

This sheet is expected by the Translator at the head of all programmes. The first line is a sentinel indicating the start of the paper tape file submitted to the Translator. The second line 'PROGN' identifies the programme. A typical entry might be:

PROGN 142 17 23 X Y 2 LTD

This would be the 23rd version of programme 142, 17. The programme numbers and version must be numeric.

The third line 'ENTRY' contains:

(i) the number of the first procedure that is to be obeyed (which need not be the first submitted).
Section 8.3

(11) priority class – an entry in the range 1 to 4 as follows:
   1 = Master Programme
   2 = Data vetting and pseudo off-line runs
   3 = Normal processing jobs
   4 = Low priority programmes

(111) spaces for the coder’s and checker’s initials (up to 5 characters in each space).

In the rest of the sheet the files used by the programme must be listed, one on each line as indicated by the notes on the form. The details given are:
   File identity (see note 3 on form) for each file
   Route type (see note 2 on form) for each file
   Chapter number, where chapter 0 = special chapter
   Maximum block size (in long words) for each file
   Alternate route required indicator
      (this is set at 1 when more than one route is required for a particular file, e.g. a multi-rewel magnetic tape file. It applies to paper tape input and magnetic tape only).

8.3 The Section Description Sheet

This sheet is expected by the Translator at the head of all programmes immediately after the program heading parameters sheet.

The first line ‘CHAPS’ specifies the number of chapters (see section 10) in this programme and the first procedure of each of them in chapter order. Chapter start points must be specified in the order in which the chapters are submitted to the Translator. Chapter order cannot be altered simply by transposing, the start points; the chapter coding must also be moved. Each chapter start point must be the first instruction of a procedure, or the first item in a table or constant section.

After the CHAPS block is an optional block STORE, which is only relevant for computers with two blocks of storage (i.e., more than 4 divisions). The entries in this block are the numbers 1 or 2, for as many positions as there are initial chapters in the programme, including the extra chapter (see section 10). The number 1 indicates first block of store, i.e., that containing the Master Programme, and the number 2 indicates the second block of store, in which the designated chapter will be preferentially allocated.

On the remainder of the sheet the details of all sections not occupied at the start of the programme must be specified, one on each line.

Amendment No. 28
November 1964

7/0
The details given for each such section are:

- Section number
- Length (in long words)
- Transit area indicator (0 = normal section, 1 = transit area)
- Chapter number (see section 10) where chapter 0 = special chapter
- Name (if any)

8.4 The Programme End Sheet

This must be the last sheet of the programme. It has three blocks, the first containing the word END, the second a sentence indicating the end of the paper tape file submitted to the Translator and the third a zero block as described in section 12.2.3.

8.5 Sheet Numbers

All sheets other than:
- Programme heading parameters sheet (see section 8.2)
- Section description sheet (see section 8.3)
- Programme end sheet (see section 8.4)
- Amendment run heading sheet (see section 15.2)
- Start and End of reel sheets (see section 8.6)

must be numbered, starting from 1, in the order in which they are to be submitted to the Translator. The numbers are entered in the space provided at the top of each sheet. The Translator will report on any data submitted out of sequence although no other action will be taken.

8.6 Start and End of Reel Sheets

A maximum of 100 programme sheets may be perforated on one reel of paper tape for submission to the Intercode Translator (see section 16). If more than one reel is required for a programme, a Start of reel sheet must be placed before the sheets for all reels except the first, and an End of reel sheet after the sheets for all reels except the last.

A 2-digit reel number must be inserted on the Start of reel sheet giving the reel number (starting at 02 for the second reel).
9. **ANNOTATION**

Up to 40 characters of annotation may be submitted against each line of the Intercode programme, with the exception of Post-Mortem Points and trial data. This annotation can comprise any characters except the control characters 'block end', 'line end' and 'alignment mark'.

Amendment No. 26
June 1964
10. **CHAPTERS**

10.1 **General**

The unit for storage allocation is the chapter, which must not be greater than one division of store (5192 compartments). The programmer is required to divide his programme into two or more chapters, if it is longer than one division, by specifying chapter start addresses; this is done on the CHAP block (see section 8.3) where a procedure number is entered corresponding to the first procedure/table/concrete section in the new chapter. Chapters are numbered sequentially from 1; and annexes and programmer's sections are specified on the appropriate sheets with a chapter number, showing into which chapter they will be translated. Pass 1 of the Translator prints out the chapter lengths, and start positions within chapter of all procedures/sections/annexes, so that the programmer may decide how to split his programme should any chapter be longer than 5192 compartments.

The programme loading sequence allows for a maximum of 7 chapters including the extra chapter; the programmer may use up to 14 chapters, where the seventh and following overlay earlier chapters (see action 152 and Appendix B to Part 2 of this volume).

10.2 **Extra Chapters**

In addition to the chapters specified by the programmer, the Translator generates an extra chapter. This holds 204 locations of allocation details for the Master Programme and the Programme Loading Sequence, followed by 20 locations for each distinct convert table used in the programme (i.e., generated by actions 63 to 65, 93), followed by all sections specified as being in chapter 0, followed in turn by all annexes specified as being in chapter 0. The Translator sets the extra chapter length at a minimum of 500 locations, since prior to the use of this space for sections/annexes, the programme loading sequence will be placed therein.

10.3 **Chapter Lengths**

In choosing chapter lengths, the programmer should bear in mind the requirements of programmes which are to time-share with his own; indeed it may be necessary to conform to the rules of the installation as to how much space may be used by particular types of programme. Since the Translator deals with inter-chapter references as interdivision references, by inserting extra instructions, a more efficient programme may result from one large chapter than several smaller ones.
A translated chapter results in computer code in which the expansion varies between 20% and 100% approximately, according to programming techniques and type of programme. Particular note should be taken of ways in which large expansions result, mainly in group 4 and group 5 actions, and certain forms of modification (see section 11 and part 2, Appendix B of this volume); it is often advantageous to place input/output sequences in a subroutine, and to use the 'absolute' method of addressing multi-record blocks (see part 2, Appendix B). An additional technique to save space taken by sections is the use of the 'dummy overlay' instruction to address the procedures at the start of a chapter as a section, e.g. when these procedures correspond to a lengthy prep stage in the programme (see Appendix B to part 2 of this volume).
11. MODIFICATION OF INSTRUCTIONS
11.1 Modification register

Intercode provides up to 20 modification registers; the group G actions, which use these, are described in section 5. Numbers 1, 2 and 3 are 'hardware' modification registers; the others are simulated by the translation process. Whenever possible modification registers 1-3 should be used.

11.2 Modification types

In the definitions of Intercode actions, modification types are given as 1 and 2. Modification of a type 1 action is more efficient generally than modification of a type 2 since in type 2 the modification has an effect to be simulated by the translator. A comparison of the times is shown below.

11.3 Modification times

The following table shows the approximate time and extra instructions required for the various types of modification. All extra instructions are inserted by the translator. A modification register with a number higher than 3 is a store compartment situated in the chapter in which it first occurs.

<table>
<thead>
<tr>
<th>Modification type</th>
<th>Modification Register Number</th>
<th>Modifies a reference to</th>
<th>Extra time in microseconds</th>
<th>Number of extra instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 - 3</td>
<td>This chapter</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1 - 3</td>
<td>Another chapter</td>
<td>52</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1 - 3</td>
<td>Transit area</td>
<td>70</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>&gt;3</td>
<td>This chapter</td>
<td>52</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>&gt;3</td>
<td>Another chapter</td>
<td>375</td>
<td>7 or 9</td>
</tr>
<tr>
<td>2</td>
<td>&gt;3</td>
<td>Transit area</td>
<td>375</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>1 - 3</td>
<td>This chapter</td>
<td>70</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1 - 3</td>
<td>Another chapter</td>
<td>325</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>1 - 3</td>
<td>Transit area</td>
<td>375</td>
<td>9</td>
</tr>
</tbody>
</table>
Section 11.4

11.4 Example

If another chapter is referred to by a modification register with a number higher than 3 which is first used in this chapter, the computer code instructions which the Translator inserts are:

0/1/0    Copy ABC
0/0/3    Set C binary
6/0/0    Parameter value for other chapter
4/0/0    Add value of modification register
2/0/0    Transfer to W/L
0/0/1    Restore ABC
24/1/2   Modify next instruction address

11.5 Notes on Negative Numbers in Modification Registers

11.5.1 The 'modify' routines of the hardware assume that the registers hold data in complement form, and negative modification is obtained by complementing the modifier with respect to bit 16 (32,768) for a 1 to 4 division machine, and bit 17 (65,536) for a 5 to 8 division machine before setting the register. The effect is to address a store cyclic within 1 to 4 or 5 to 8 divisions, such that division 0 in division 4, etc.

The Translator will complement negative modifiers with respect to bit 17 in the case of the 90 and 93 actions, but the programmer must do this for actions 91 and 92. The extra bit 16 in the complement does not affect addressing on a 1 to 4 division computer.

11.5.2 To ensure correct testing for end in the 94 and 97 actions, if the modifier and end values are of different sign, or both negative, each must have bit 16 or 17 added when the modification register is first set up. This is done automatically by the Translator for action 90, by adding bit 17, but the programmer must do this for action 91.

11.5.3 If by accident a negative modifier is set in the register without first complementing, the number -x will in general behave as x in a modified action. In certain cases, attempts to obey the modified action will cause lockout. Further, if such a register is stepped, overflow will occur.

11.5.4 Negative step values are held in modulus form, with bit 13 of the instruction address functioning as a sign bit. Hence no step value may numerically exceed 4095. Bit 13 (4096) is added to the modulus of the step value by the Translator for action 94, but must be done by the programmer for action 97.

Amendment No. 26
June 1964
Section 12

12. FILE CONTROL

12.1 General

File control is carried out by the Master Program under the direction of the program file control actions (group 4, see section 5). The use of the Master Program in this way means that the program must obey certain rules; these rules are stated below since they are binding on the use of the intercode file control actions. No justification for them is included since they are fully dealt with in Volume IV of the Users' Manual which describes the Master Program.

The medium of the file is described by a route type on the Program Reading Sheet. The following is a list of available route types, which do not necessarily all exist on a given installation; supplementary descriptions are given later in the section.

<table>
<thead>
<tr>
<th>Route Type</th>
<th>Medium</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7-hole PT output, in a 6-bit code with odd parity inserted by the hardware in a fixed position (see 13.5)</td>
</tr>
<tr>
<td>2</td>
<td>Non-standard 80-column card output (see 12.5)</td>
</tr>
<tr>
<td>3</td>
<td>Standard (decimal) 60-column card output (see 13.3)</td>
</tr>
<tr>
<td>4</td>
<td>Anlex printer on Standard Assembler (see 13.6, 14.1)</td>
</tr>
<tr>
<td>5</td>
<td>5 or 7-hole PT input (see 12.7, 13.4)</td>
</tr>
<tr>
<td>6</td>
<td>80-column (decimal) card input</td>
</tr>
<tr>
<td>7</td>
<td>Magnetic tape, first Assembler (see 13.1)</td>
</tr>
<tr>
<td>8</td>
<td>Magnetic tape, second Assembler (behaves as 7 if only 1 assembler)</td>
</tr>
<tr>
<td>9</td>
<td>Anlex printer on special Assembler (see 13.6)</td>
</tr>
<tr>
<td>10</td>
<td>80-column non-standard card input (see 12.6)</td>
</tr>
<tr>
<td>11</td>
<td>Magnetic tape, third Assembler (behaves as 7 if only 1 or 2 assemblers)</td>
</tr>
<tr>
<td>12</td>
<td>Autolector (see 12.13)</td>
</tr>
<tr>
<td>13</td>
<td>40-column card input (Upcine reader) (see 12.12)</td>
</tr>
<tr>
<td>14</td>
<td>Special form of dual printer (see 19) for software use. All file control orders must be in computer code.</td>
</tr>
<tr>
<td>15</td>
<td>Dual input (PT or cards) (see 12.9)</td>
</tr>
<tr>
<td>16</td>
<td>Non-standard PT Output (see 12.6)</td>
</tr>
<tr>
<td>17</td>
<td>Cheque Sorter (see 12.13)</td>
</tr>
<tr>
<td>18</td>
<td>Non-standard 40-column card input (see 12.12)</td>
</tr>
<tr>
<td>19</td>
<td>Dual printer i.e. special or standard assembler (see 12.10)</td>
</tr>
<tr>
<td>20</td>
<td>Data Transmission Link input (see 12.14)</td>
</tr>
<tr>
<td>22</td>
<td>Data Transmission Link output (see 12.15) Amendment No.40 June 1967</td>
</tr>
</tbody>
</table>
12.2 Sentinel Blocks

12.2.1 Computer-Prepared Files

At the head of each output file (except printer files) the Master Programme writes a 'Start of File' sentinel block containing the identity of the file. On input this block as a whole is not made available to the programme, although some of the information it contains is left in register 4 after opening a file (see note vi to section 40).

At the conclusion of each output file (except printer files) the Master Programme writes an 'End of File' sentinel block. On input this block is not made available to the programme.

On magnetic tape only, the Master Programme writes another standard block before the End of File sentinel. This further block is made available to the programme, and contains:

Word 1 : Serial numbers as usual (see section 13)
Word 2 : 5 alignment mark characters
Word 3 : 'FINIS'
Word 4 : Block end word (see section 13)

12.2.2 A Card Input File must include Start of File and End of File sentinels. These are placed automatically on computer-produced files (see section 12.2.1) but on other files they must be prepared manually. The layout of the sentinels is given in section 12.2.4.
12.2.3 **A Paper-Tape Input File** must include:

i. Start of File sentinel (at start of first reel).

ii. Start of Reel sentinels (at start of all reels except the first).

iii. End of Reel sentinels (at the end of all reels except the last). This must be followed by a zero block.

iv. End of File sentinel (at end of last reel). This must be followed by a zero block.

These are placed automatically on computer-produced files (see section 12.2.1) but on other files they must be prepared manually. The layout of the sentinels is given in section 12.2.4.

12.2.4 **Sentinel Block Layouts on Paper Tape and Cards**

<table>
<thead>
<tr>
<th>Characters</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 5</td>
<td>Each has control 7 and basic 1. This forms a sentinel indicator, so that no data block may have its first five characters in this form.</td>
</tr>
<tr>
<td>6 - 8</td>
<td>Identity number of programme suite.</td>
</tr>
<tr>
<td>9 - 10</td>
<td>For computer-produced files, identity number within suite of programmes which produced the file. For other files any characters.</td>
</tr>
<tr>
<td>11</td>
<td>File identity letter.</td>
</tr>
<tr>
<td>12</td>
<td>For computer-produced files, file identity number, for other files, any character.</td>
</tr>
<tr>
<td>13</td>
<td>Sentinel type:</td>
</tr>
<tr>
<td></td>
<td>1 = Start of reel or file</td>
</tr>
<tr>
<td></td>
<td>2 = End of reel (paper tape only)</td>
</tr>
<tr>
<td></td>
<td>3 = End of file</td>
</tr>
<tr>
<td>14 - 15</td>
<td>Paper tape: Reel number (starting at 0) This need not be specified on end of reel or file sentinels.</td>
</tr>
<tr>
<td>Cards</td>
<td>Spaces</td>
</tr>
<tr>
<td>16</td>
<td>Paper tape: Block end character</td>
</tr>
<tr>
<td>Cards</td>
<td>Space</td>
</tr>
</tbody>
</table>

Remaining columns on Spaces card
12.3 Input and Output Notes

12.3.1 Parity Checks

For media other than cards, parity is checked by the equipment and action taken, if necessary, by the Master Programme.

When a parity error is discovered on paper tape input the block must be re-read so that the operator must re-align the file manually. As a check on re-alignment the operator aligns the file to read in the last correctly read block. The Master Programme maintains a record of the first two words of the last correctly read block and uses this to check the re-alignment.

In order to strengthen this check, the programmer should ensure, if possible, that the first two words uniquely identify the block.

12.3.2 Serial Number Checks

The Master Programme provides a block serial number test on magnetic tape. Any similar check required on other media (e.g. punched cards) must be provided by the programme.

12.3.3 Blank Cards

A card Input File should not contain completely 'blank' cards. Any blank cards will be ignored (i.e. not made available to the programme).

12.3.4 Printer Form Length

For 'on-line' printing using route type 9 forms should not exceed one block in length.

12.3.5 Time of Closing Input File

On input files, if the Close File instruction is given when the 'End of File' sentinel is not the next block, a comment is made in the log and the file closed. The programme continues.
Section 12.4

12.4 Re-alignment

12.4.1 Rewind (Action 45)

A file which has been rewound must be opened before further use.

Following a rewind, a file may be opened as an input or an output file. Checks are performed to ensure that the same file is concerned as was opened initially.

12.4.2 Accuracy of Run Back (Action 46)

The accuracy of the 'run back' action is not guaranteed when the alignment marks are within eighty words of the block end word of the block containing the alignment marks. In such circumstances additional time may be taken as the Master Program aligns the tape to the specified block, due to its failing to find the marks at the first attempt.

12.4.3 Restart Facilities

The Master Program does not provide any restart facilities. To enable programs to provide these if required, the facility to change the function of a file by running to alignment marks is provided, but this facility must not be used to replace block(s) in the middle of a file.

12.5 Non-Standard Card Output

This is used to punch cards other than in the format available with decimal card output, e.g., in a binary code for another computer or with triple punching in a column (cf. section 15.3). The file is specified as route type 2 on the file description sheet, and the output order assumes that the programmer has set up the bit pattern required by the hardware in his output section.

The medium is handled exactly as for decimal cards, by the Master Program; thus an open file order causes a start sentinel to be punched, and similarly for a close file order.

Output is from a 26-word section, where no special character is needed to indicate the end of the information. A full description of the hardware requirements is given in Volume I, Section 15.4.2, of which the following is an abbreviation:

The first and second words of the section are to be clear.

Amendment No. 3/41
March 1968.
The 80 bits comprising the third, fourth words correspond to the 80 columns in the top row of the card, i.e. the 'tens' row; the presence of a bit indicates a hole is to be punched, where bit 1 of the third word = column 1, i.e. LHS of card, bit 2 = column 2, etc.

The fifth, sixth words define the second row, and so on.
Note:  
(i) These words must all be positive.
(ii) To avoid difficulty in punching/reading a weakened card, programmers should not cause more than half the card to be used, e.g. alternate columns should be blank.

12.6 Non-Standard Card Input

This is used to read cards which contain multiple punching in a column, beyond that needed to cover the LEO III alpha code; for example, to read non-standard card output, or cards produced by another computer. The file is specified as route type 'O' on the file description sheet; this will cause an even route number to be allocated to the programme, where in fact the card reader is connected simultaneously to an even and odd route (usually 4, 5 or 6, 7). An open file order will read the sentinel in the standard (decimal) mode, on the even route, and initiate the reading of the first card of the file in the same mode. Thus the programme's first read order will make available this card, which is normally some form of run heading card; this read order also initiates the reading of the second card of the file in the non-standard mode, on the corresponding odd route number. Subsequent read orders will call the odd route number. The close file order will cause an EDO comment as the sentinel is expected by the Master Programme to be in decimal format (cf. section 12.3.5).

The layout of a section containing a non-standard card is as follows:

Word 0' = columns 1, 2 (24 bits) with column 1 in bits 18 to 29, row 10 = bit 18 and column 2 in bits 30, 31 ... 40, 52 with row 10 = bit 30.

Word 2' = columns 3, 4 etc.

Word 78' = columns 79, 80.

Bits 1 to 17 of the words will be cleared by the assembler. Note that the form of the input section is not suitable for non-standard card output.

12.7 Non-Standard Paper Tape Input

No special provisions are made by the Translator or Master Programme to distinguish non-standard from standard paper tape. A key on the PT reader desk causes each read order to read one word into the store ignoring parity checks.

The 'open file' (and associated promote read) is omitted; the programme's first read order will make available a blank section, and the second read order, the first 5-character word in the most significant end of the first word of the section. Similarly, the xth read order makes available the (x-1)th word. The close file order is also omitted.
12.8 Non-Standard Paper Tape Output

On installations where paper tape is punched with one machine code output/action per character, i.e., using the General Purpose Assembler (the alternative arrangement outputs 5 characters at a time), file control orders for route type 16 (non-standard PT O/P) function as follows:

(i) open file: tests route available, but punches no sentinel.
(ii) close file: sets route to manual, without punching a sentinel.
(iii) output: punches the given block, terminated by a block end character, but does not output the block end character itself.

This facility is used for punching in a 7-hole code different from LRD III alpha code, and all file start/reel end/reel start/file end sentinels are punched by the programmer using the normal output facility. 6-bit characters are punched as specified with odd parity inserted by the hardware. There is a facility to detect warning of end, and the punching of a doubtful character, both described in note (iii) to the 42 action description.

12.9 Dual Input (Paper Tape/Card)

This is route type 15, and the allocation routine will choose either a paper tape route or an 80-column card route according to the following logic:

(i) Test Master Programme indicator set for 'preferred input = PT'
   If so, attempt to allocate a paper tape route; if no paper tape route available, allocate card; if neither available, reject allocation.

(ii) If the indicator is set for 'preferred input = card', attempt to allocate a card route; if no card route is available, allocate paper tape; if neither available, reject allocation.

The indicator is normally fixed for each installation's Master Programme although certain Master Programmes may allow a command to change it at run time.

The 'open file' action 40 will leave (B) = 5 if paper tape was allocated and 6 if card was allocated. The 42, 43 actions work normally, and the difference in unpacking or interpreting the data is handled by the programmer.

Note: All file control actions for route type 15 must be in an initial chapter, which is never subsequently overlayed into the store again.

Amendment No. 25
November 1964
12.10 Dual Printer

This is route type 19, designed so that without programme amendment the programme may be run using either the special or the standard Amellex printer (cf. route types 4 and 9). Multi-line blocks can be printed on either type of printer by this method. A programme can contain not more than two files of route type 19.

The columns on the Programme Heading Parameters Sheet are used in a non-standard fashion, as follows:

<table>
<thead>
<tr>
<th>Column</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Identity</td>
<td>(as required)</td>
</tr>
<tr>
<td>Route Type</td>
<td>19</td>
</tr>
<tr>
<td>Chapter of Annexe</td>
<td>Two decimal digits, of which either or both can be zero (see below for these cases). Indicates the chapter position (not Intercode chapter number) into which the transit area and annexes respectively will be translated.</td>
</tr>
<tr>
<td>Block Size</td>
<td>Block length for standard printer, a multiple of 64 (i.e., as if route type 4).</td>
</tr>
<tr>
<td>Alternate Route Indicator</td>
<td>Block length for special printer, i.e., as if route type 9.</td>
</tr>
</tbody>
</table>

As an extra transit area is generated in the programme, whose length will be zero if a special printer is allocated or a multiple of 64 words (the same as the printer annex) if a standard printer is allocated.

Special Printer

The position of the annex will be at the end of the indicated chapter, following the print subroutine (see below) if this is the extra chapter. If the indicated chapter is zero, the annex is treated temporarily as a further chapter, and eventually placed at the end of some existing chapter, but so as to give the most favourable allocation of storage.

Standard Printer

The positions of the two annexes will be at the end of their indicated chapters, following the print subroutine if the chapter is the extra chapter. If one or both indicated chapters are zero, one or two temporary further chapters are created; allocation then proceeds as described in the preceding paragraph.
Print Subroutine

The expansion of the 43 action consists of an entry to a subroutine in the extra chapter, with parameters. This subroutine occupies 16 locations if a special printer is allocated or 50 locations if a standard printer is allocated. The subroutine is placed in the extra chapter after all 'normal' annexes or sections, but before the print annexes.

It should be noted that the chapter sizes stated in the machine code summary of a program, do not make allowance for the print subroutine or standard printer annexes. This should therefore be taken into consideration when determining how much space a program will occupy in store.

12.11 Autolocator

This is route type '12 which reads input document data straight into program. All information whether of doubtful block condition or not is read into the transit area. The programmer must deal with the doubtful block condition by using a continuation line with his input action. If no doubtful block link is present 2222 will occur. Two input areas must be specified. The close file action sets route to manual.

12.12 40-Column (Uptime) Card Reader

The major difference between this reader and the 50-col reader, is that on the uptime reader there are two output hoppers so that a primary sort can be made while reading. These are used as 'normal' and 'reject' hoppers. The choice is made by an 'output' instruction for each card read; also, any card for which the 'output' instruction arrives too late is automatically directed into the 'reject' hopper.

12.12.1 Action 40 Open File

Behaves as for 50-col card input, the Master Program directing the sentinel card into the 'reject' hopper and initiating the reading of the first data card. Normal sentinel checks are made.
12.12.2 Action 42 Read Card
Behaves as for 80-Col card input.

12.12.3 Action 43 Output Card
The output section number in the first C-line is to be the same as for the associated read order. Register A is to contain zero at the start of the action to address the 'reject' hopper, and to contain 1 to address the 'normal' hopper. The second C-line contains a procedure start address to which control is passed (binary radix registers A and B 'indeterminate') whenever the card arrives in the 'reject' hopper, either because it was meant to or because the 43 action came too late; the programmer should verify the choice. The maximum permitted delay is about 15 milliseconds.

12.12.4 Action 41 Close File
Behaves as for 80-Col card input. The Master Program directs the end sentinel into the 'reject' hopper.

12.12.5 Non-standard Mode
Actions 40 and 42 behave as for 80-Col card input, see section 12.6. The other actions behave as for the Standard mode.

12.13 MICR Cheque Sort Set/Reader
This peripheral is controlled by Intercode actions 132, 133, 134, and 135, described below. It reads data at a constant rate, with an input 'pipeline' of three cheques. The computer code 'read' order causes a data transfer for the current cheque and also maintains the data flow; if the read order is late, cheques in the pipeline are automatically diverted to the 'reject' pocket, and if it is absent for a sufficient time the data flow will stop. After reading a cheque can be directed into one of 13 pockets, automatically going into the 'reject' pocket if the computer code 'pocket select' order is too late.

The medium is treated as single-axial (unbuffered) input. Following a read order, the Master Program delays return to the cheque sorter program until the read senses to become engaged, without obeying other programs in the priority queue. The computer is in the non-interruptible mode between return to program after the read order and entry to Master for the 'pocket select' order, so that no delays occur here due to time-sharing.
Any program time-sharing with the cheque sorter program should have lower priority, and have no channels in common with it (so as to avoid error conditions which will hold up both programs); in particular, the cheque sorter program should have a 1/2 assembler to itself.

12.15.1 Action 132 - Open File

The cheque sorter is given a conventional file identity and an annex of nominal length 1 word. The purpose of the 132 action is to test device ready, and set up the time annex for the sorter to coincide with the stated input section.

Action : 132
Reference : file identity
Item : input section number (transit area)

N.B. No other input/output device must use this section number in a file control action.

12.15.2 Action 133 - Read

The action is used for the first and all subsequent cheques. According to the setting of the literal, the data transfer terminates at one of two special symbols in the data and the time to perform the action varies. The input section is cleared in the action expansion, before the read order. The action has a return address in its o-line, to which control is passed as follows:

a) one of the characters in the cheque was doubtful and has been replaced by the quartet value 14, with the word containing it set negative; control is returned to the first instruction of the indicated procedure with binary radix set and registers A, B indeterminate.

b) one of the cheques in the pipeline is a "check end" ticket, and all cheques should now be tested for the end sign (otherwise the test can be omitted for speed); control is returned to the second instruction of the indicated procedure, with binary radix set and registers A, B indeterminate.

Action : 133
Reference : file identity
Item : input section number (as for the 132 action)
Section 12.13.2 (Cont'd)

Literal : 0 = terminate data transfer at 'serial end' symbol
          1 = terminate data transfer at 'sort code end' symbol

O-line address : procedure start address (see above) for detection of unusual conditions.

For the three possible exits from the action, the computer remains in the non-interruptible mode. The appropriate 'pocket select' order must in all cases now be obeyed before any other file control order.

12.13.3 Action 134 - Pocket Select

The required pocket is indicated by Q1 of the first word of the input/output section, Q2-10 being zero. This takes values 0-11 for the 12 normal pockets and 15 for the 'reject' pocket. The action has a return address in its o-line, to which control is passed if the 134 action came too late (the cheque will be automatically placed in the 'reject' hopper), with binary radix and registers A, B indeterminate.

Action : 134
Reference : file identity
Item : section number as used for the 132, 133 actions
o-line address : procedure start address for detecting late pocket select

12.13.4 Action 135 - Step Batch Counter

This action is not yet available.

12.13.5 Form of Data in the Store

Cheques pass from left to right under a read head, and the characters are read as quater values 0-9, with 4 special symbols 10-13 and the 'doubtful' value 14. Data transfer into a 10-quarter buffer register begins when the first special symbol is read, which is also treated as part of the data. The first character appears in the Q1 position, the next in the Q2 position, etc. The buffer register is transferred to the next store address whenever another special symbol is read, this appearing both at the most significant end of the current word and at the least significant end of the next word. Data transfer stops on the recognition of one of two special symbols.
Section 12.14

Data Transmission Link Input:

Single annex (unbuffered) input is used. The file annex is given a nominal length of one word, and is not used for data transfer. All data is read into one specified transit area; this section must be the only one referred to in file control actions for this file and must not be used in file control actions for any other file.

The file control actions are I/O actions 40 and 42, and, if desired, a second C-line to the read action may be used to specify a procedure number to which control will be returned in the event of the detection of a "doubtful block" condition.

Data Transmission Link output:

Normal double annex working is used. The file control instructions are identical to those for normal non-standard card output, (section 5). However a special C-line to the output word is mandatory in this case. Control will be returned to the specified procedure in the event of the detection of a doubtful block. If no return link is specified the Master Program will close the program, giving a 2200) comment, and the program must be abandoned.
13. **BLOCK LAYOUTS**

13.1 **Magnetic Tape**

13.1.1 The first two words of each block will, on output, be over-written by the Master Program; on input the first two words will contain information written by the Master Program. The first word contains, in fact, the serial number of the block within the reel, the second contains a count of the number of doubtful blocks that occurred on trying to write this block.

13.1.2 **Block End Word**

At the end of each block for output there must be a word containing 65 - 6 zero quartets - 6B. This word is written on the tape and read back into the store on input. After input the word need not be replaced before subsequent output.

13.2 **Alpha/Numeric Punched Card Input**

13.2.1 Information is read into the first word of the block (i.e. the first word is not left clear). The maximum block size quoted in the program heading parameters sheet must be at least 16 words. (Reasons for quoting a block size greater than 16 are given in section 14).

13.2.2 The first word of the block must not contain five ‘/‘ characters (i.e. control quartet of 7 and basic quartet of ’), unless the block is a ‘Sentinel’.

13.3 **Alpha/Numeric Punched Card Output**

The first word of the block is ignored on output. The first character of the second word is punched in the first column of the card. A block end character may be inserted anywhere in the block, other than in the first word, and will cause output to cease, although it will not be punched. If a block end character is not inserted earlier, it must be placed in the 8th character position (in this context the first character is the most significant octet of the second word). Only numeric characters 0-11, alphabetic characters A - Z, and space may be output.
13.4 Paper Tape Input

13.4.1 The first word of the block must not contain:
   a. five '1' characters (i.e. control quartet of 7 and basic quartet of 1), unless the block is a sentinel.
   b. five '0' characters (i.e. control quartet of 5 and basic quartet of 0). This pattern in the first word is used as an indication that parity failure occurred on output and has not been corrected. In these circumstances the program is unloaded.

13.4.2 Each block must conclude with a 'block end' character.

13.5 Paper Tape Output

Each block must conclude with a 'block end' character.

13.6 High Speed Printer

The first word of each line must be a 'function' word, as follows:

<table>
<thead>
<tr>
<th>Function</th>
<th>Represented by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero line feed</td>
<td>B17 (facility not available for route type 4)</td>
</tr>
<tr>
<td>1 line feed</td>
<td>Spaces</td>
</tr>
<tr>
<td>Move paper to next hole on plastic tape in track 1</td>
<td>B1</td>
</tr>
<tr>
<td>2</td>
<td>B2</td>
</tr>
<tr>
<td>3</td>
<td>B3</td>
</tr>
<tr>
<td>4</td>
<td>B4</td>
</tr>
<tr>
<td>5</td>
<td>B9</td>
</tr>
<tr>
<td>6</td>
<td>B10</td>
</tr>
<tr>
<td>7</td>
<td>B11</td>
</tr>
<tr>
<td>8</td>
<td>B12</td>
</tr>
</tbody>
</table>

Amendment No. 3/41
March 1966.
More than one of these bits may be present, in which case the first hole reached in any of the tracks indicated stops the paper movement.

In addition, for printing using route type 4 the following bits of the function word are used:

<table>
<thead>
<tr>
<th>Function</th>
<th>Represented by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print space characters in positions 1 - 40</td>
<td>B37</td>
</tr>
<tr>
<td>41 - 80</td>
<td>B38</td>
</tr>
<tr>
<td>81 - 120</td>
<td>B39</td>
</tr>
<tr>
<td>121 - 160</td>
<td>B40</td>
</tr>
</tbody>
</table>

Any combination of these bits may be set by the programme. Bit 40 is automatically set by the translator for installation having a 120 character printer. The spaces are additional to any put in the output section by the programmer, e.g. if bit 38 only is set, then the first 40 characters after the function word will be printed as characters 1 - 40, the next 40 as characters 81 - 120, and so on.

For files using route type 4, each block consists of one line only which must conclude with a 'line end' or 'block end' character occurring on or before the 161st character (not including the function word). This character is not printed.

If the 'line end' or 'block end' occurs earlier than the 161st character, all characters after the last information character, up to and including character 40, 80, 120 or 160 (whichever comes next) are printed as spaces. Unless the appropriate bit of the function word is set, subsequent characters will be the same as those on the previous line but one. This is important if mixed long and short lines are being printed.
(13.6) For files using route type 9, each block may consist of any number of lines. Each line must conclude with a 'line end' character, except the last line which must be terminated by a RE character. When a LE character occurs in a word, all following characters in that word are ignored and the next word is treated as the function word of the next line. The most significant character in the word after this will be the first character printed in the following line.

Care should be exercised that the block end does not occur in a word which will be treated as a function word in multi-line blocks.
14. TRANSIT AREAS

14.1 Definition

Transit Areas are those areas of the store used for input or output of data. There are two types of transit area, annexes and input/output sections:

Annexes

One annexe is created by the Translator for each file with the exception of card output files and printers on the Standard Assembler (route types 3, 4). For these, two annexes are created which have the following lengths:

route type 3. 2 annexes each of \((25 + 2)\) words
route type 4. 2 annexes each of \((192 + 2)\) words (120 col. printer)
2 annexes each of \((256 + 2)\) words (160 col. printer)

The block length of printers on route type 4 specifies the GP
annexe required:

- 64 words columns 1 - 40
- 128 words columns 1 - 80
- 192 words columns 1 - 120
- 256 words columns 1 - 160

All other annexes have the length specified on the File
Description sheet with 2 words added.

Annexes are placed in the programme's special chapter inserted
by the Translator, and are not directly addressable by Intercode
actions. Of the pair of annexes required for route types 3 and 4,
the first appears as an 'input/output section' in the Pass 3 heading
results and the second as an annexe; although both are situated in
the special chapter.

Input/Output sections

These areas are allocated for each section indicated as a
transit area on the Section Description Sheet, and for which an
input or output instruction exists in the Intercode programme.
The exception is a section used solely for output on a file of
route type 3 or 4, as orders are inserted into the object program
to move the data from this section to one of the pair of annexes
before the output instruction is obeyed. The length of the area
allocated is the length as given on the section description sheet
plus two words. An input/output section is not directly accessible
to the peripheral equipment.

Amendment No. 19
October 1963
Section 14.2

The Master Programme keeps a list of the starting addresses of all transit areas. Data can then be 'transferred' between annexes and input/output sections by interchanging the starting addresses in this list, thereby interchanging the functions of the areas. In the case of route types 3 and 4, however, data is physically transferred from output section to the two annexes alternately. Rules for using transit areas are given in sections 14.2 to 14.6 below.

14.2 Input and Output instructions may make reference to any input or output section, and one section may be used for input or output on several files.

14.3 Input into specified section will overwrite the previous contents.

Output from a specified section will corrupt the contents of that section (i.e., successive output instructions must not be given from the same section unless data is placed in that section after each instruction), except for files of route type 3 or 4.

14.4 Total Area Required by Annexes

Subject to the rule in section 14.5 the total area required by annexes is the sum of:

\[ \text{maximum block size quoted for file on programme} \times \text{words} \]

for all files except those having route types 3 or 4; and

\[ \text{Annex area as specified in section 14.1} \times \text{words} \]

for all files having route types 3 or 4.

This total area must not exceed 3,966 words.

The Master Programme uses the programme transit areas in order to read/write sentinel information. This means that annexes and input/output sections must have the following minimum lengths:

- Magnetic Tape 10 words
- Paper Tape 8 words
14.5 Sets of transit areas

An annex or input/output section is said to belong to a certain set of transit areas if data is transferred between that annex or input/output section and any other member or members of the set by interchanging functions of areas. (Note that a transit area can only belong to one set).

The two annexes used for files of route type 3 and 4 are not included in the set with the associated output section, since data is here transferred physically.

The section length or maximum block size quoted in the programme heading information must be the same for all members of the set and must be the largest of the individual values.

An example of the effect of this rule is given in section 14.6.
Example of store space required for transit areas

Consider a programme with:

<table>
<thead>
<tr>
<th>Input Files</th>
<th>with maximum block size (words)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A6</td>
<td>40</td>
</tr>
<tr>
<td>B1</td>
<td>100</td>
</tr>
<tr>
<td>C1</td>
<td>50</td>
</tr>
<tr>
<td>D5</td>
<td>20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0</td>
</tr>
<tr>
<td>I0</td>
</tr>
<tr>
<td>D0</td>
</tr>
</tbody>
</table>

The programme uses the following input/output sections:

- **SECTION 7** acts as both input section for file A6 and output section for file AO.
- **SECTION 8** is used as input section for D5.
- **SECTION 12** is used as output section for D0.
- **SECTION 13** is used as input section for B1 and C1 and as output section for AO.
- **SECTION 14** is used as input section for C1.

The relationship between input/output sections and files may be depicted as:

```
A6                  AO
(40)                (40)
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>SECTION 7</td>
</tr>
<tr>
<td>1-----------------</td>
</tr>
<tr>
<td>B1</td>
</tr>
<tr>
<td>(100)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
```

It will be seen that there are 4 'nets' of transit areas.
1. Annexes for files A5, A0; Section 7. It follows from section 14.5 that the length quoted for each member of this set must be 40 words.

2. Annexes for files B1, C1, D0; Sections 13, 14. The length quoted for each member of this set must be 100 words.

3. Annex for file D3; Section 8. The length quoted for each member of this set must be 20 words.

4. Annex for file D0; Section 12. The length quoted for each member of this set must be 20 words.

<table>
<thead>
<tr>
<th>SECTION or ANNEX for file</th>
<th>APPARENT LENGTH</th>
<th>LENGTH WHICH MUST BE QUOTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>A5</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>B1</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>C1</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>D3</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>A0</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>D0</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>D0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>12</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>13</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>14</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>Total</td>
<td>420</td>
<td></td>
</tr>
</tbody>
</table>

Total storage space required for annexes is thus (420 + 14) words = 434 words (see section 14.4).

An addition (290 + 10) words i.e. 300 words is required for input/output sections. This area may be divided by the programmer between chapters according to the available space.
15. PROGRAMME AMENDMENTS

15.1 Introduction

When a programme is amended, the Translator reads the Intercode programme file produced at the last run for this programme in parallel with amendments submitted on Paper Tape/Cards. The amendments are identified either by a data type (for the programme heading information) or by the line serial held on the programme file against the data it is required to amend. The amendments are to be submitted in ascending line serial order, preceded by an amendment heading block, and followed by the end sign as for an initial translation.

Specimens of the data sheets used are given in Appendices C and D.

15.2 Amendment Heading Block

This specifies the suite, job and serial of the programme to be amended; following this is the trial indicator, the printout indicator, and the new issue number of the programme.

Trial Indicator

This is either blank, or the letter 'T'. Blank indicates that the PHS programme is not to be written into the programme tape, i.e. the programme cannot be trial allocated (this does not prevent the programme's containing trial data, or its being amended this run). 'T' indicates that the PHS programme is to be written onto the output tape.

Printout Indicator

This is a combined indicator controlling both the amount of Inter-

code and the amount of computer code printed.

0 = reduced Intercode, reduced computer code
1 = reduced Intercode, full computer code
2 = full Intercode, reduced computer code
3 = full Intercode, full computer code

The full Intercode printout consists of the entire programme held on the file, and is obtained automatically at an initial translation. The reduced Intercode printout includes the programme heading with the files, chapter and section description, and subsequently is governed by the following conditions:

(i) if any procedure, table or constants section is amended, the complete procedure is printed. This includes procedures inserted in the current amendment. All incoming and internal sequence changes are printed.
Section 15.3

(i) if the number of the procedure alters as a consequence of the insertion/deletion of procedures elsewhere in the programme, the procedure is printed, together with incoming and internal sequence changes.

(iii) all reported lines are printed, if not included in (i) or (ii).

(iv) whenever the address of an instruction or table entry alters as a result of insertions/deletions elsewhere in the program although this particular line has not been amended, the entire procedure containing the line is printed.

(v) the post-mortem points are always printed.

(vi) the entire trial data set, in which there is an amendment, is printed.

The reduced computer code printout consists of the heading information followed by the section start, location of switches/item + counters/indirect modification registers, followed by the procedure start address (12 per line). The full printout displays the contents of each location on the programme tape with its parameter, by procedure.

New Issue Number

The value submitted here, if non-zero, replaces the top 3 digits of the brought forward programme serial. The two least significant digits of the serial are stepped by 1 at each amendment, but return to 0 when 99 is stepped.

15.3 Amendment of Programme Heading Information

The Programme Heading Parameters Sheet, incorporating the file description, is amended by re-submitting the sheet; by this means file identities, number of files, entry point, programme identity, etc are all taken from the new sheet.

The STONE block can be replaced or inserted, but not deleted.

The chapter structure of the programme is amended by replacing the CHAPS block. The section description is amended according to the method given in 15.5.

15.4 Amendments of Instructions, Table Entries and Constants

15.4.1 General

Each instruction, table entry and constant is given a serial number during Pass 1 of the Translator and items are amended by reference to this serial number which will have been printed by the Translator during Pass 2. The first three digits of the serial number are the Translator's reference number and the last two digits are the Translator's item number (see section 16.3). There are four types of amendment, as described below.

Amendment No. 31
May 1969
15.4.2 To Replace Items Starting at Serial Number S

The serial number of the first item to be replaced is entered in the serial number column of the amendment sheet followed by an 'R'. The amendment is then written as for normal coding beginning at the same line as the directive. Translator's reference and item numbers must be used (see section 15.4.1). The first of each set of replaced constants must be accompanied by a constant directive, i.e. A, B, C, D, B or P.

If the NOTES block is being replaced, then a fresh 'replace' directive must be given for both this line and the next replaced line in the procedure.

15.4.3 To Delete Items with Serials 81 to 82 inclusive

The two serial numbers are entered on the amendment sheet, the first being followed by a 'D' in the serial number column. If the second line does not occur, a single item is deleted. No entries are required in other columns.

If the deletion begins in one procedure and ends in another, then the amendment must be submitted as a pair of deletions, the second of which must start at the beginning of the second procedure. For example, to delete from 21766 to 21807 where the last line of procedure 217 is 21754 the amendment must be submitted as a pair of deletions 21766/21754 and 21805/21807.

A NOTES block can be deleted as a single line deletion, but a deletion 'from...to...' must not begin at a NOTES block.

In the special case where a procedure contains too many entries as the result of previous amendment, so that there are several lines with line serial xxx98 (table or long constants) or xxx99 (instructions or short constants), then a deletion of the form xxxxx/xxx98(99) will delete up to and including the first line with serial xxx98(99). The single deletion xxx98(99)n will delete all lines with that serial.

15.4.4 To Insert Items Before Item of Serial Number S

The serial number is entered and followed by a 'I'. The inserted item(s) are then entered, starting on the next line. The first of each set of inserted constants must be accompanied by a constant directive, i.e. A, B, C, D, B or P. To enable references to be made to items with the insertion, these may be given five-digit serial numbers by the programmer.

Each such serial number must be unique within the programme. The first three digits of such a number must be different from the first three digits of any serial number existing before amendment. As with the 'Replace' facility above, Translator's reference and item numbers must be used throughout.

Amendment No. 29
December 1964
These unique serial numbers should only be used if it is necessary to refer by an instruction to an item within an insertion, and not more than 50 may be used on any one amendment run. If a unique reference has been used within an insertion to address a long location, the serial plus one cannot be used to address the odd half. If insertions are made at the end of an existing procedure they must be addressed as if they were continuations of the procedure, without the use of unique references. Any other current insertions or deletions to this procedure are to be ignored in predicting such addresses. The odd half of a constant or table entry inserted in this way can be addressed by the predicted (odd) serial number.

A NOTES block must not be inserted before the first line of a TABLE or a PROCEDURE. The block is not essential for correct translation, and can safely be omitted.

15.4.5 To Replace Annotation of Instructions Starting at Serial Number 3

The serial number is entered and followed by an 'A'. Only the annotation is changed and this is completely replaced. The annotation is to occur in the same position in a block as it would for a 'replace' directive, i.e. indented by the appropriate number of 'number end' characters. The facility can be used to amend section description, directive blocks PROC/TABLE/CONV/NOTES, and constants, instructions or table entries.

When amending annotation on a directive block the directive PROC/TABLE/CONV and the procedure number must be left blank.

If the annotation of NOTES block is being replaced, then a fresh directive must be given for both this line and the next replaced line in the procedure.
15.4.6 Amendments Affecting Procedure Headings

(i) Insertion of a New Procedure

The directive block PROC/ TABLE/ COND is indented by one number and, e.g. it may be written in the 'action' column for the PROC block, with the procedure number written in the 'reference' column, and the annotation beginning in the 'item' column.

The number chosen for the new procedure must be different from all procedure numbers in the brought forward program, and any other procedures being inserted.

Lines in the inserted procedure must be addressed without the use of the 'Unique Serial' facility (see 15.4.4), by using the number of the inserted procedure and an item within procedure beginning at 02 for the first line, increasing by 1 for instructions/short constants and by 2 for table entries/long constants.

The 'P' facility may be used for internal sequence changes, instead of the procedure number.

Note: A new procedure may only be inserted before the start of another procedure, or before the END directive.

(ii) Deletion of a Procedure

If the directive line is deleted, the Translator will assume that the entire procedure has been deleted and will report any lines addressing that procedure. Any remaining data in that procedure will be amalgamated with the previous procedure; but this practice is to be avoided as reference to the amalgamated data (e.g. sequence change, or addressing a constant) will not be adjusted for the new position of the data.

(iii) Replacement of a Procedure Heading

A TABLE type may be altered by replacing the directive block, but no other amendment is acceptable; the procedure number cannot be changed, and a short constants section cannot be changed into a long constants section and vice versa.

8/9 Amendment No 3/41
March 1968.
15.4.7 Note on Re-allocation of Serial Numbers and Adjustments made to Addresses

(a) In the case of insertions or deletions within a procedure, table, or constant section, new serial numbers will be allotted to subsequent items within this procedure, table, or section.

The following adjustments are made by the Translator to addresses of instructions:

(i) Any instruction or table entry addressing a deleted item should have its address amended in the current amendment run; the Translator will comment if this is not done, since in this case the resultant address may not be acceptable to the programmer. However, the address printed with the comment will be the actual value used in calculations.

(ii) Any instruction or table entry referring to an item which occurs after an insertion or deletion in a procedure, table or constants section has its address adjusted to refer to the new position of that item.

No adjustments are made by the Translator to relative address constants; the programmer must amend these if necessary.

(b) If a complete procedure, table or constants section is inserted or deleted, all subsequent serial numbers will have their first three digits changed. Any references to subsequent items by instructions or table entries will be adjusted by the Translator to address the new positions of these items. The procedure number (but not the item) of relative address constants is adjusted by the Translator. The printout in Pass 2 will contain the procedure before adjustment but the translation will use the adjusted procedure. Note that unique serial numbers cannot be used as relative address constants.

15.5 Amendments to the Section Description

Each line of the section description is given a serial number during Pass 1 of the Translator, consisting of an asterisk followed by a 4-digit number. This number takes values 1001 by increments of 1 to a maximum of 1099 (corresponding to 99 possible section numbers).

The method described in 15.4.2 to 15.4.4 is used to replace/delete/insert sections, where the serial number specified on the editing sheet includes the asterisk; e.g. to replace the first section, the directive is *1001R. It is not possible to insert a new section before the first procedure start; however, the insertion can be made before the final section, which is also included in the insertion, and simultaneously deleting the line containing the final section.

The directive 'replace annotation' is acceptable for amending section description.

Amendment No. 24
May 1964
16. **TRIAL FACILITIES**

This section should be read in conjunction with the 08005 program specification (Volume V of the Users' Manual) and the Program Trial System Specification (Volume IV of the Users' Manual).

16.1 **Program Trial System**

This is a sequence of orders written onto the program tape, and held in the computer when this program is trial allocated, fulfilling two main purposes:

(i) To set up input magnetic tape files at run time, from data held on the program tape and amendable by the Translator or 08005, thus removing the need to reserve tapes or to write data generator programs.

(ii) To allow intermediate and final states of the program's store area to be written onto a magnetic tape, for display at the end of the program under trial, or as a separate run (this decision depends upon the installation performing the trial).

16.2 **P2S in written onto the program tape for an initial translation if a TRAND directive is submitted or if an indicator is submitted in the AMEND block (see section 15.2), and only read into the computer if the program is 'trial allocated' (see Volume IV); when used, it causes the program's storage requirements to be increased by 1500 locations.

16.3 **Trial Data**

This is the data held on the program tape to be copied at run time onto the program's input tapes. A full description of the data submitted to the Translator for inserting, replacing or deleting this information is given in the 08005 program specification.

16.4 **Post-Mortem Points**

These are points within the program at which the contents of storage are to be dumped, the program then continuing as normal. The programmer indicates up to 20 potential points by writing TRANJ in the first five characters of annotation of an instruction, and then indicates a particular choice by data submitted to the Translator or 08005 quoting the Inter-code line serial of the line containing TRANJ.

Amendment No. 3/41
March 1968.
Section 16.5

Use of the Translator to Amend Trial Data or Post-Mortem Points

Data is submitted as for 08005, the heading block TRAMD replacing the Intercode end sign END and following all Intercode amendments. If required, the 08005 data file can be fed directly to the Translator, thus by-passing the main translation routines but automatically adding PCS to the output program tape.
17. THE OPERATIONAL PROGRAMME

17.1 Programme File

The Translator output magnetic tape file contains the Intercode programme, followed by the Programme Loading Sequence, followed in turn by the computer code programme.

17.2 Allocation

When a programme is to be run, a command is given to the Master Programme which causes it to read a control block (see Volume IV, Part I, section 1); this control block specifies the programme identity and serial, and causes the Master Programme to search the programme file for the heading block placed at the start of the loading sequence for the programme. The first part of the loading sequence is read in by the Master and entered. Control remains with the loading sequence until the programme is in the store.

17.3 Programme Loading Sequence

This is a standard sequence of orders generated by Pass 3 of the Translator before the computer code programme. It consists of six parts, interspersed with the allocation requirements of the programme as follows:

(i) Unloader routine - to update the 'facilities available' tables for any programmes in the store which have reached end or been abandoned by command.
(ii) 'Allocate storage' routine - to decide into which parts of the store the programme shall be loaded.
(iii) Block giving all allocation requirements except routes, transit areas.
(iv) 'Allocate routes, transit areas' routine.
(v) Up to three blocks giving routes, transit areas requirements.
(vi) 'Allocate floating point/mod. group/tag' routine.
(vii) 'Update facilities table/form programme parameters' routine.
(viii) 'Load programme' routine.
(ix) Computer code programme.

17.4 Parameters

In the operational programme as it is presented to the Master Programme, each short word has associated with it a 'parameter' occupying the next short word.

Amendment No. 28
November 1964