ELLIOTT 903

Volume 2: PROGRAMMING INFORMATION

Part 2: PROGRAM DESCRIPTIONS

Section 3: T. 2 (TRANSLATION INPUT ROUTINE)

Contents

Chapter 1: DESCRIPTION

1.1 INTRODUCTION

1.1.1 Purpose

T. 2 (translation input routine) is used to read, translate and place in store programs or other data written in machine-code.

1.1.2 Versions

The standard version is stored immediately below location 8180 and is distributed as a binary tape for input by the initial instructions. A relocatable version is available and distributed as a machine-code program for input by T. 2 or the Symbolic Input Routine (SIR).

1.1.3 Method of Use

Three entry points are provided to allow for first and subsequent tapes of a program (see sub-paragraph 1.1.6).

T. 2 may run in any program level. It stores the program it is translating in the module it itself occupies.

1.1.4 Compatibility

Programs written according to the conventions of 920 T. 2 are acceptable to 903 T. 2.

1.1.5 Glossary

In the following glossary, references are given to the paragraphs in this section where the terms are explained more fully.

Block (1.2.2.1)

A section of program to be placed by T. 2 in consecutive store-locations. During translation T. 2 places each word it translates into the current block.

Block Address (1.2.2.1)

The absolute address of the first location of a block.

Block Number (1. 2. 2. 1)

The position of the block address in the directory.

Constant (1. 2. 5)

This term is used to describe a location used in a program as a constant: it may be punched as an integer, a fraction, or as a pseudo-instruction.

Directory (1. 2. 2. 1)

A list of the block addresses to be used by T. 2 when translating a program.

Instruction (1. 2. 1. 1 and 1. 2. 5. 1)

An instruction consists of;

a function
a separator
and an address.

Newline Sequence (1. 2. 1)

In this section <newline> refers to the sequence of characters:-
<carriage return> <linefeed> <null>.
T. 2 ignores <carriage return> and <null>, <linefeed> is the significant character. The full sequence should be punched to obtain a good print-up.

Word (1. 2. 1 and 1. 2. 5)

A word is a section of program to be placed in one store location. It may be
an instruction
an integer
or a fraction.

1.1.6 Entry Points

(Absolute addresses refer to the standard version of T.2, while relative addresses refer to the relocatable version).

To translate the first tape of a program	enter at 8 or 7749 or 0;
To translate subsequent tapes after a wait	enter at 9 or 7906 or 1;
To translate subsequent tapes commencing a new block	enter at 10 or 7850 or 2;

NOTE: The entry-points 8, 9 and 10 are provided for the convenience of the user. Under certain circumstances they may be corrupted and entry must then be made at the alternative address.

1.2 FUNCTIONS

The program tape input by T.2 contains:

(1) A sequence of words, each terminated by <newline>(described in 1.2.1.)

(2) Characters which act as directives to T.2 concerning the placing of the words on the program tape (these are described in 1.2.2.).

1.2.1 Words Translated by T.2

A word may have one of three formats: it is terminated by <newline> and placed in one store location by T.2.

1.2.1.1 Instructions

An instruction is punched as:-

(1) A function written as a decimal number from 0 to 15 inclusive, which may be preceded by /.

(2) A separator.

(3) A group of characters specifying the address.

(4) <newline>

Function - If the instruction is modified the function digit(s) is preceded by / e.g. /8 1

Separator - one, and only one, space must appear between the function and the address.

Address - This may one of the following:-

(1) A decimal number from 0 to 8191 (inclusive) representing the absolute address of a store location. e. g. 8 8177

(2) A decimal number from 0 to 8191 (inclusive) further defining a 14 or 15 instruction. e. g. 15 6144

(3) A decimal number followed by; which represents an address relative to the current block address.
e. g. 4 10;
refers to the address "10 more than the current block address".

(4) Two decimal numbers separated by ;
e. g. 8 1;10
refers to the address "1 more than the address of block 10".

NOTES (1) All numbers in an instruction are unsigned decimal integers.

(2) The first block has number 1. The first location of a block has relative address 0;

(3) It is recommended that instructions are punched as shown in the examples above.

1. 2. 1. 2 Integers

An integer must be preceded by + or - and be left justified. It may contain from one to six (inclusive) decimal digits. It is terminated by <newline> and must not contain a decimal point.

The range of values, n, that can be translated is given by

$$|n| \leq 131071$$

e. g. +131071
-123

1.2.1.3 Fractions

A fraction must be preceded by + or - followed by one decimal point. It may contain from one to six (inclusive) decimal digits, terminated by <newline>. The range of values, x, that can be translated is given by

$$|x| < 1.0$$

e.g. +.999999
+.5

Fractions are stored as unrounded binary numbers: the number stored may have a maximum error of 2^{-17} (0.0000076).

1.2.2 Directives

1.2.2.1 Blocks

A block consists of a sequence of words each terminated by <newline> which are placed by T.2 into consecutive store locations. Each block is terminated by * followed by <newline>.

The absolute address of the first word of the block is called the block address and is specified by the programmer in the directory. The first block is numbered 1.

The directory is punched at the beginning of the program. It must have the following format with each line terminated exactly by one <newline>.

Character Punched	Significance
&	This is a permanent directory and defines the addresses of the following blocks. The block count becomes zero.
+200	The address of block 1 is 200
+4000	The address of block 2 is 4000
+1700	The address of block 3 is 1700
	Note that these must be signed integers although they represent absolute addresses.
*	The block count is increased to one and T.2 begins to translate and store words of program.
	If the coding is on a separate tape the directory should be terminated by an end-of-tape symbol only.

The block addresses are used during translation to

(1) Place new blocks.

(2) Calculate relative addresses (see 1. 2. 1. 1).

1. 2. 2. 2 End-of-tape Symbol

When <halt> is read T. 2 waits. This allows the tape to be unloaded from the reader, and another tape translated (at the second entry to T. 2).

Note that a 903 library tape must be translated as a new block.

1. 2. 2. 3 Title

A title may be punched on a tape to identify it. The title is preceded by = and terminated by <null>. All characters between are copied by T. 2 on the punch as they are read. A title may be punched only on the first tape of a program and must be the first item on the tape.

1. 2. 2. 4 Temporary Directory

This facility may be used for editing programs in store or placing parameters in a program already translated.

The temporary directory is a signed address and is followed by a group of words to be stored in consecutive locations from the address specified.

The address and group of words may be preceded by a title: except for this they must be the only items on a single tape. The tape is translated by entering at 8.

The directory refers to the main directory already translated: e. g. +97;1 instructs T. 2 to begin storing at the address "97 more than the first block address in the main directory. " The current block number becomes equal to 1 and the current block address equal to the first block address.

The words are punched according to the normal T. 2 conventions: they are separated from the directory only by <newline>. Addresses may be relative to any block address already translated.

Characters Punched	Significance
+1;3	Begin storing at the address "1 more than the third block address in the last permanent directory read"
+14	+14 is stored in location 1;3
8 1;	The current block number is 3. Therefore 8 1; is translated as 8 1;3 using the third block address as for the first example and stored in location 2;3.
14 1	14 1 is stored in location 3;3
8 10;2	8 10;2 is translated using the second block address in the last permanent directory read and is stored in location 4;3.
£	£ causes the computer to enter a dynamic stop.

The directory may be punched as an absolute address: e.g. +113

The group of words is then treated as a separate block with the directory as the current block address. The block count is not preserved if an absolute directory is used.

1. 2. 3 Ignorable Characters

The following characters are ignored when read by T 2, subject to the qualifications stated:-

<null> - ignored everywhere except when terminating a title.

<delete> - ignored everywhere.

<carriage return> - ignored everywhere.

<linefeed> - ignored if no meaningful character read since the last <linefeed> or since the last entry to T. 2, except when a directory is being translated. In a directory an extra <linefeed> is an error.

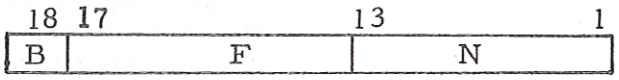
1. 2. 4 Errors Detected.

Certain error conditions are detected by T. 2 (see Paragraph 1. 3).

1. 2. 5 Binary Pattern Stored.

1. 2. 5. 1 Instructions.

A location (18 bits) is divided as follows:



Bit 18=1 if the instruction is modified.
=0 otherwise.
Bits 17-13 give the function as a binary number.
Bits 13-1 give the absolute address for the instruction:
i. e. addresses relative to the start of
a block are stored as absolute addresses.

1. 2. 5. 2 Integers.

These are stored as binary numbers.

Bit i represents 2^{i-1} for i = 1, 2 . . . , 17
Bit 18 " the sign of the number.

Negative numbers are held as a complement relative to 2^{18}
e. g. +4 is held as 000 000 000 000 000 100
-4 is held as 111 111 111 111 111 100

1. 2. 5. 3 Fractions.

These are stored as unrounded binary numbers.

Bit i represents 2^{i-18} for i = 1, 2, . . . , 17
Bit 18 " the sign of the number.

Negative numbers are held as a complement relative to 2. 0

e. g +. 5 is held as 010 000 000 000 000 000
-. 5 is held as 110 000 000 000 000 000

Since the number is unrounded, the number stored may be less than the correct value, but by not more than 2^{-17} i. e. 0. 000 007 6.

Note that most patterns may be stored in one of three ways, subject to the limitation of the inaccuracy of storage of fractions.

e.g. 010 000 000 000 001 000

may be stored by punching 8<space> 8 (Instruction)

+ 65544 (Integer)

or + .500 061 (Fraction)

Some patterns may be stored only by using an instruction.

e.g. to store - 131072

or - 1.0

punch - /0<space>0.

1.2.5.4 Locations Used for Storage

Each word, as defined in 1.2.1, is stored in one location: all words in one block are stored in consecutive locations from the absolute address specified in the directory.

If locations are to be reserved as workspace this may be done by specifying the contents as +0. If large areas of store are to be reserved a separate block can be used, and the directory used to specify the number of locations. It will be necessary to punch only

* <newline>
* <newline>

on the program tape. The contents of these locations will remain unaltered during translation.

If T.2 is over-written this may lead to an error indication.

1.3 ERROR INDICATIONS

If any of the errors below are detected, then the character

01111.111

is output continuously. If this occurs, stop the computer and examine the characters near the point where reading stopped to discover the error. Translation must be restarted when the program tape is corrected.

The errors detected by T.2 are as follows:-

(1) Impermissible character input.

(2) Contextual error.

(3) Format error.

(4) Directory error: a directory address is greater than +7740.

(5) Overwriting error: the last block stored has overwritten T. 2.

Tests (4) and (5) are not included in the relocatable version.

1.4 METHOD USED

1.4.1 Directory Store.

In the standard version, up to 85 block addresses may be used. In the relocatable version workspace is reserved for up to 47 block addresses. Block addresses are stored in consecutive locations at the end of the program; therefore the reserved area may be extended by the user in the relocatable version.

1.4.2 Block Count.

The current block count is set to zero when & is input at the head of the main directory. It is increased by one when * is input and when T.2 is entered at 10. The use of an absolute temporary directory also increases the count by one.

1.5 STORE USED.

The standard version uses locations 7740-8179 (inclusive).

The relocatable version uses 400 consecutive locations.

1.6 TIME TAKEN.

T.2 operates at the speed of the tape reader.