

Digital Computer Laboratory
211 Massachusetts Avenue
Cambridge 39, Massachusetts

To: S&EC Group Staff Members

From: J. M. Frankovich

Date: March 15, 1955

SUBJECT: INTERIM REPORT ON THE WWL-ERA 1103 INPUT TRANSLATION PROGRAM *

Introduction:

The facilities of automatic coding systems can be made available at different levels both of complexity and sophistication. The compiler-interpreter developed by Al Roberts and described in earlier Newsletters, for example, facilitates the assembly of programs coded in a pseudo-code, where pseudo-coded instructions refer to lower level coded routines. A basic system, less sophisticated but no less desirable, is represented by the Ramo-Wooldridge Assembly Program. Such a system simplifies the problems inherent in coding and later checking and modifying long sequences of machine coded routines.

The WWL-ERA 1103 Input Translation Program written by members of the Digital Computer Laboratory at MIT is also intended to simplify the machine coding problem. It will allow programmers for the ERA 1103 to code their programs in a mnemonic (two letter) operation code and to use either or both symbolic and integer addresses. The translation program accepts Flexo-coded punched paper tape as input and produces a seventh-level bi-octal punched paper tape acceptable as input to the 1103 computer. The program operates on the Whirlwind I computer at MIT and was written under the sponsorship of the Digital Computer Laboratory and Project DIC 7138 at the Servomechanisms Laboratory at MIT. It will be used in the coding of a large data reduction program presently being developed at the Servomechanisms Laboratory and expected to be run on the Eglin Field 1103. However, anyone who desires to explore the facilities offered by this translation program may do so by writing to the S&EC Group at the Digital Computer Laboratory.

* To be published in the ERA 1103 Newsletter

Description:

The following is a brief summary of the vocabulary and syntax of the translation program:

A. Characters

Standard Flexowriter equipment is to be used in the preparation of the punched paper tapes. All lower case alphabetical and numerical characters are distinguished by the translation program except for a deliberate ambiguity between "o" and "0". No upper case letters are allowed except in titles or comment words (See F. 1 and 3). The space, color shift, and stop characters are ignored, as are combinations of shift to upper case and shift to lower case characters which do not affect the appearance of a program print. The back space and code combinations for which no Flexo characters have been assigned are considered illegal. The tab and carriage return are synonymous.

The words in the vocabulary of the translation program will, for the most part, be composed of syllables of letters and digits suitably punctuated and terminated.

B. Words

Three classes of words are defined. The first class consists of all those words which occupy storage registers in the translated program. These words are instructions, with zero to four addresses, or (integer) numbers. This class of words we will polysyllabic storage words.

The second class, called polysyllabic control words, consists of current address assignments, symbolic address assignments, and starting address assignments. These words influence the form, location, and operation of the translated program but do not themselves occupy registers of storage.

The third class of words, called special words, are used for miscellaneous control and identifying purposes. They are the title, the number base indicator, and the comment word.

The words of the first two classes are combinations of these syllables: operations, symbolic address tags, integers and the literal addresses "q", "a" and "b". These words are distinguished solely by the internal and terminal punctuation. These characters are "+", "-",

"=", ",", ".", " | ", the tab, and the carriage return.

C. Syllables

1. Operations. These are the mnemonic lower case two-letter pairs corresponding to the standard listing of the 1103 operation code.

2. Symbolic address tags. These are three character digit and letter combinations. All are included in the following listing: a01, a02, ..., a99, b01, ..., z99; lal, ..., 9al, la2, ..., 9a9, lbl, ..., 9z9; 0la, ..., 99z. The letter "o" is not allowed in any of the tags, so slightly less than 7500 tags are available.

3. Literal addresses. The letters "q", "a" and "b" will have the values 10000, 20000, and 30000 octal, respectively.

4. Integers. Integers may have any value up to $2^{35}-1$, so long as the completely evaluated word, or part of a word, of which the integer is a syllable, fits into the number of bits meaningfully available in the translated word. Integer syllables are always converted to binary from the base specified by the most recent base indicator special word (See F.2), appearing in the program, (decimal if no base is specified).

D. Polysyllabic Storage Words

1. Instructions. An instruction word always has the operation code as the initial syllable. Zero to four addresses may follow; the translation program will decide where to position the addresses, bit-wise, in the word if a meaningful number of addresses is given. The addresses are separated by commas and the last is terminated by a tab or carriage return. Each address is written as the sum of symbolic address tags, literal addresses, and integers, where the manner of summation is indicated by prefixing each syllable of the address by a plus or minus sign.

Examples are: (" | " indicates a tab or carriage return)

tp lal+5,q74 |

ef 37 |

rp 3,23,30h |

ms l,a |

2. Numbers. A number has the form of an address of an instruction word and is terminated by a tab or carriage return. The value of the translated number must be less than $2^{35}-1$.

Examples:

40₂
1a1+b7-4q+60₂

E. Polysyllabic Control Words

1. Current address assignments. A current address assignment has the form of an address of an instruction word except that here the terminating character is a vertical bar: "|". This word causes the next polysyllabic storage word to occupy the register of 1103 storage, high-speed or drum, whose address is the value specified in the current address assignment. Successive storage words will thereafter go into successive registers of storage until another current address assignment occurs.

Examples:

100|
1b2|
60+1a7-q9|

2. Symbolic address assignments. This word consists of a single symbolic address tag and is terminated by a comma: ",". The value assigned to the tag is the location which a polysyllabic storage word would occupy if one were the next word on the Flexo tape; this is true whether or not one occurs there. A tag can be reassigned several times in a program, the last value being used in the translated program. All reassigned tags are indicated as such by the translation program.

Examples:

1a1,
b7,

A word which has the form of an address of an instruction word, which does not consist of a single symbolic address tag, and which is terminated by a comma will be evaluated and compared with the location of the following storage word. A disagreement will be indicated by the translation program.

Examples:

5+1a1,
50+b7.

3. Starting address assignment. A starting address assignment work must occur at the end of each Flexowriter program tape. It has the form of a single address instruction word, where the place of the operation

code is taken by the syllable: "start st". The value of the starting address indicates where operation of the translated program is to begin.

Examples:

start at 40
start at lal

F. Special Words

1. Titles. A title is used for identifying Flexo tapes and logging the translation of the tapes on Whirlwind I. A title must have the following form:

f2r ddd-ddd-ddddd xxx...xxx

The d's represent the digits of the identifying tape number, and the x's any additional information. The title must be terminated by a carriage return. A title must appear at the beginning of each tape to be translated.

2. Number base indicator. This word is written as "base k", where k is any integer from one to ten. All following integer syllables in the program will be converted to binary from this base, until another base indicator occurs.

3. Comment word. This word is totally ignored by the translation program and is provided so that comments may appear on a program print when a program tape is printed. This word must have a vertical bar occurring initially and is terminated by a tab or carriage return. Any Flexo character may occur in between.

Examples:

tp lal,a | Jump to q74
ej b7,q74 | if(q74) = (lal)

G. Symbolic Addresses

This translation program has the distinctive ability to do the following:

If a programmer decides that he needs to modify a section of his program and does not desire to reproduce the entire Flexo-tape in order to make the required deletion and insertion, then he need only append a tape of the modified section to the main tape and have both translated together. The translation program will then effectively insert the appended section of the program into the program, expanding

or contracting the original section of the program to occupy the same number of storage locations as the replacement. The facility is thus available to insert or delete words in a program by making modifications at the end of the tape.

This facility is available, however, only for those parts of programs which are labelled by symbolic addresses, for addresses which are specified by their integer values cannot be relocated during the translation process. For example, suppose we are given the following program (the dashes represent storage words):

```

1a1, -
      -
      -
      -
      -
      -
      -
      -
      -
      -
4q5, -
      -
      -
      -
      -
      -
2a3, -
      -
      -

```

and we wish to replace the eight words between 1a1 and 2a3 by six words and reposition 4q5. Then at the end of this original program tape we would attach, for example, the following modification:

```

1a1  -
      -
      -
      -
      -
      -
4q5, -
      -
      -
      -
      -
      -
2a8, -
      -
      -

```

The word originally labelled by 2a3, and all following words (until perhaps a current address assignment occurred) would be moved up two registers and all symbolic addresses involved would be reassigned to have a consistent value. The six words beginning at 1a1 are replaced by the six new words of the modification and the third, rather than the fourth, after the one labelled by 1a1 is now labelled by 4q5.

At the end of each program translation all the values for all the symbolic addresses assigned in the program are listed. All unassigned and incorrectly assigned symbolic addresses are also listed. Hence a programmer has a complete record of the storage locations used by his program, as well as of the effect of modifications on the program.

One comment should be made about the structure of the symbolic address tags. At the 18 January meeting at MIT the decision was made that the tags would have the forms: digit-digit-letter or letter-digit-letter. In this translation program the not quite equivalent forms: digit-digit-letter, digit-letter-digit, or letter-digit-digit are allowed. If the translation program is revised at all, it will include a more general form for tags which will allow all these tags to be used.

H. Error Detection

Illegal combinations of characters are detected, and so recorded, by the translation program wherever possible. The locations of the errors are given in terms of the most recent symbolic address or current address assignment.

I. Separate and Simultaneous Translations

The translation program is a two pass system. As many tapes as desired, within the limitation imposed by the number of available symbolic addresses, may be read in together by the translation program during the first pass, and then punched out together during the second pass. Such program tapes may have cross references to one another and are said to have been translated simultaneously. On the other hand, two tapes, or groups of tapes, may have been separately run through both the first and second pass, in which case there can be no cross references. Such tapes are said to have been separately translated.

J. Prospective Addition to the Vocabulary

1. Numbers. Integer syllables in number words will shortly be allowed to have the following more general form:

$$\pm 123.456 \times 2^{\pm 5} \times 10^{-6}$$

where the rounded off integer part of the resulting number is taken as the integer syllable in the number word.

2. Interpret. No decision has yet been made as to how to handle the address part of an interpret instruction, but it will probably be made to depend upon the needs of the particular program being translated.