RECOMP II USERS' PROGRAM NO 1067

PROGRAM TITLE:	BINARY TABLE LOOKUP ("BTLU") RECOMP SUBROUTINE (RELOCATABLE)
PROGRAM CLASSIFICATION:	General
AUTHOR:	T. B. Feigenbaum Autonetics
PURPOSE:	To find the address in memory at which is stored a given piece of information
DATE:	28 March 1961

Published by

RECOMP Users' Library

at

AUTONETICS INDUSTRIAL PRODUCTS

A DIVISION OF NORTH AMERICAN AVIATION, INC.
3400 E. 70 Street, Long Beach 5, California

BINARY TABLE LOOKUP ("BTLU")
RECOMP SUBROUTINE (RELOCATABLE)

1. PURPOSE

1.1 To find the address in memory at which is stored a given piece of information.

2. RESTRICTIONS

2.1 Table arguments must be stored in ascending order (-2, -1, 0, 1, 2, etc.) in consecutive memory locations. The first argument may be in any location MMMM.0; the second must be stored in MMMM.0 plus 1.0; the nth in MMMM.0 plus (n-1).0.

2.2 At the beginning of a search, the Search Argument must be in the A Register at the same scale as the table arguments.

3. METHOD

3.1 The routine uses the "binary search" method of table lookup. It looks first at the mid-point of the table; if the Search Argument is larger than the mid-point, the next test is of the word halfway between the mid-point and the upper limit. This procedure continues, with the size of the table being successively divided by two, until the Search Argument is found in the table, or is found to be absent from the table.

3.2 The speed of this method can be illustrated by the fact that a "worst case" can be found in a 1000-word table in eleven passes through the search loop--all entirely within the high-speed loops. For a 2000-word table, only one additional pass would be required.

4. USAGE

4.1 Linkage Instructions:

a. Place the Search Limits in XX26 and the Argument Extractor in XX27 (see Definition of Terms).

b. Place the Search Argument in the A Register at the same scale as the Table Arguments.

c. Transfer to the Subroutine.

4.2 Definition of Terms:

a. Search Limits - The four-digit addresses of the first and last words in the table. These appear as a thirteen-digit octal word in the form "+00000UUUULLLL" where UUUU is the address of the last word--the upper limit of the search--and LLLL of the first word, the lower limit

4.2 Definition of Terms: (continued)

b. Argument Extractor - A mask which will extract the Argument from a word in the table and set all other bits to zero. If, for example, the Argument appears in the nine low-order bits and is assumed to be negative, the Extractor (in command format) would be "-000000.0 - 000377.1."

4.3 Result of Routine:

At the conclusion of the routine, the A Register will contain one of the following:

a. If the Search Argument was found in the table, the word "+000000.0 - 00AAAA.0" where AAAA is the address in the table where the Search Argument was found.

b. If the Search Argument was not found in the table, the word "-000000.0 ± 00AAAA.0" where AAAA is the address of the Table Argument last examined (the sign of the right-hand half word is not significant). This means that the Search Argument was larger than the Table Argument at AAAA-1 and smaller than the Table Argument at AAAA+1.

4.4 Suggestion:

The restriction at 2.1 does not restrict each element in a table to one Recomp word. Table Functions may be so stored that the address in memory of a Table Function is a function of the address of the Table Argument. For example, a table of 128 fixed-point Arguments could be stored at 1000-1177 with the corresponding floating-point functions at 2000-2376. In this case, the address of the Function is double the address of the Argument.

4.5 Storage Required: 48 full words (XX00 to XX57 inclusive) of which the last eight are used for temporary storage.

4.6 This subroutine is relocatable. The master tape contains AN-004, and is subject to the restrictions of that program.