

RECOMP II USERS' PROGRAM NO. 1080

PROGRAM TITLE: MEMORY SEARCH

PROGRAM CLASSIFICATION: Utility

AUTHOR: W. Wellman
Servomechanisms, Inc.
Research Division
Goleta, California

PURPOSE: This routine searches all of memory (except for this program's instructions) to find any specified instruction and/or address.

DATE: 12 June 1961

Published by

RECOMP Users' Library

at

AUTONETICS INDUSTRIAL PRODUCTS

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

PURPOSE:

This routine searches all of memory (except for this program's instructions) to find any specified instruction and/or address.

METHOD:

1. Half words of memory are brought into the accumulator individually, in sequential order; bits extracted by the specified mask; and the result compared with the given instruction. If the masked half-word matches this instruction exactly, its location and entire contents are typed out.

2. Examples showing use of the "mask":

(a) To search for all STO 0312.0 instructions, enter:

+60 0312.0 +77 7777 0. This mask indicates that the half-word bit is of no interest.

(b) To search for all CFV instructions with a channel 20 address, enter:

+65 2000 0 +77 7700 0. This mask causes the sector bits of all addresses to be ignored.

(c) To search for all CFV and CFL instructions with a channel 20 address, enter: +65 2000 0 +75 7700 0. This is similar to (b) but now the instruction "2-bit" will also be ignored, so that the routine will search for both +65 and +67 instruction codes.

USE:

1. Go to 7730.0, the location of this routine.
2. In command fill, enter the instruction for which the routine is to search and the mask which it is to use, all in one word, in that order, as shown in the

examples above.

3. Set typewriter "tab/C. R." switch to "C. R.", and push "start" on the console.
4. The routine will now search through memory, printing the locations of all such instructions and the instruction pair in each such location.
5. After all of memory has been checked, 8 blanks will be typed, the display cleared, and the computer halted in 7730.0, waiting for a new instruction-mask pair, (Go back to step 2 to instigate a search for something else.)

RESTRICTIONS & RANGE:

1. In the interests of speed, this routine ignores the signs of all instructions; the signs of the code word parts are also ignored, and may be either plus or minus.
2. All half-words from 0000.1 through 7730.0 are checked.
3. The typewriter C. R./ tab switch must be set for "C. R."
4. Storage: this routine uses both high-speed loops and main memory locations 7730 through 7757.
5. Time: Approximately 80 seconds for all checking, plus 5.6 seconds for each printing.
6. HTR (± 77) instructions will be located correctly, but in this case extraneous printing may occur.

Program No. 107 Title MEMORY SEARCH

Programmed by: W. Wellman

Date 2-61

Loc'n	Cm'd	Addr.	Contents	Accumulator	b	Remarks
7.7.3.0			INSTRUCTION			
			MASK			
1	+66	77.50.0				
	+02	77.3.0.1		- inst, mask		
2	+41	00.2.4.0		- mask in left		
	+60	77.7.7.0				
3	+40	00.2.4.0		- mask in right		
	+60	77.7.3.0				
4	+02	77.3.0.1				
	+33	77.7.7.0	left mask	- inst in left		
5	+60	77.7.2.0				
	+40	00.2.4.0		- inst in right		
6	+60	77.7.6.0				
	+64	77.4.0.0				
7	+36	77.3.0.0				
	+57	77.6.0.1				
7.7.4.0	-77	00.4.7.0	initial count			
	*+3.0	00.0.0.0	initial location	W contents		(W+1) contents in R
1	+33	77.7.3.0	right mask	W.1 contents		
	+03	77.7.6.0	right inst			
2	+50	77.4.7.0				compare same →
	+00	77.6.0.0		W		
3	+01	77.7.5.0	bump	bump		
	+53	77.5.6.0				overflow: W=7731 →
4	+60	77.6.0.0		set new W		
	+43	00.0.0.0		contents of new W		
5	+33	77.7.7.0	left mask	W.0 contents		
	+03	77.7.2.0	left inst.			
6	+50	77.4.7.0				compare same →
	+57	77.6.0.1				go again → 7740.1
7	+72	00.0.1.0.0		type CR		← here to print
	+72	00.3.3.0		✓ figs		
7.7.5.0	+0.0	77.6.0.1		W		
	+42	77.7.4.1				
1	+42	77.5.2.1				
	+12	77.7.4.0		type W		
2	+72	00.0.4.0		✓ space		
	+12	00.0.0.0.0	W	✓ contents of W		
3	+72	00.0.4.0		✓ space		
	+57	77.6.2.1				bump W + continue*
4	+0.0	00.0.0.0.0				
	-0.0	00.0.0.0.0	zero			.1 * R = W+1 or
5	-0.0	00.0.0.1.0				.0 R = HTR(W+50)
	-0.0	00.0.0.1.0				
6	+0.2	77.5.4.1		blank		
	+72	77.6.0.0				NOISE MARKER
7	+36	77.5.4.0				CLEAR DISPLAY
	+77	77.3.0.0				→ START

*OPTIMUM FOR ALL BUT SECTORS X0, X1 ; 10₁₀ disc rev for 8 sectors = 80 rev./CHANNEL