MINITRACE 3 9/10/63

Emanuel Melichar
Economist
Board of Governors of the Federal Reserve System
Washington 25, D.C.

PURPOSE

Minitrace 3 is a trace program for IBM 1401 computers with 4,000 digits or less of core storage that are not equipped with the Advanced Programming feature (index rogisters and store register instructions). (Minitrace 1 or Minitrace 2 must be used on computers equipped with the Advanced Programming feature.) It is designed to furnish the information normally provided by a full trace program while using a minimum of core space and requiring almost no set-up work. A trace program is used in testing and debugging other programs, its output consisting of information about the execution of the program being traced. Minitrace 3 monitors each instruction of the program being traced, and except for chained instructions prints one line for each such instruction after it is executed. The contents of this line are as follows:

Print positions	Information printed
1- 3	Address of the instruction being traced.
4	The letter "C" if one or more chained instructions follow the instruction being traced.
6	Op-code of the instruction being traced.
8-10	A-operand of the instruction being traced.
12-14	B-operand of the instruction being traced.
16	d-character of the instruction being traced.
30-43	Up to 14 digits of the contents, after execution, of the field addressed by the A-operand of the instruction being traced. An asterisk indicates the length of the field.
45-58	Up to 14 digits of the contents, after execution, of the field addressed by the B-operand of the instruction being traced. An asterisk indicates the length of the field.
59	Always a blank.
60	Always a record-mark.

For a detailed description of the use and output of the Minitrace programs, see "The Trace Method for Debugging Programs on the IBM 1401," Newsletter, Committee on Computers in Research, Federal Reserve System, August 23, 1963

MACHINE REQUIREMENTS

Minitrace 3 uses 1124 digits of core memory and may be assembled anywhere in core except the print area. For example, it may be assembled with origin at 2876 to use positions 2876-3999.

LIMITATIONS

Minitrace 3 will trace all generally known 1401 instructions except those included in the Advanced Programming feature. There are several minor limitations and requirements, fully described in the Write-Up, that are imposed in order to reduce the core space used by Minitrace 3. The author would like to be contacted if a case that cannot be traced is encountered in order that the program may be modified to handle the situation.

OPERATING PROCEDURE

Minitrace 3 has been designed to avoid almost all "setting-up" of the program deck to be traced. In most cases, it is merely necessary to remove the "END" card from the deck to be traced, place Minitrace 3 behind the remainder of the deck, place any data cards used behind Minitrace 3, and load and run the combined deck, following exactly the procedures specified for the program being traced.

Procedure for trace of entire program

1. Remove the last card (END card) from the program deck to be traced.

2. Place the Minitrace 3 deck behind the program deck to be traced.

3. Place data cards, if any, behind the Minitrace 3 deck.

4. Load and run the combined deck.

   Follow the procedures specified for the program being traced.
   (Check for switches, carriage control tape, etc.)

Precaution--If not altered, Minitrace 3 will expect to find the first instruction of the program being traced in core location 333. To begin tracing at a different address, that address should be inserted as the contents of FX, the constant that occupies the first 3 digits of Minitrace 3. For example, if Minitrace 3 is assembled with origin at 2876 and the program to be traced starts at V01, then V01 should be inserted into 2876-2878.

REQUIREMENTS, LIMITATIONS, AND FURTHER EXPLANATION

A. Chained instructions

Minitrace 3 can accomodate up to 31 digits of consecutive chained instructions in the program being traced. In other words, if the program to be traced does not contain a string of more than 31 consecutive digits of fully-chained and/or half-chained instructions, it can be traced.

B. Incorrect indication of presence of chained instructions.

If, in the program being traced, it happens that one-digit constants that contain characters that are op-codes that can be chained follow an unconditional branch instruction in core storage layout, Minitrace 3 will be fooled into thinking that a chain will be executed, and will print "C" after the address of the branch instruction. This incorrect notation does not otherwise affect the trace, and is mentioned here only because it might be confusing in the rare occasions when it is encountered. A false notation will also be printed if a four-digit constant containing an L or M in the high order position follows the unconditional branch.

C. Explanation of use of asterisks to indicate length of fields

When the contents of the fields addressed by the A-operand and B-operand are moved prior to being printed, the move is stopped either by the word-mark of the field addressed or when the 14th digit of the field is moved. The asterisk is then printed in the position immediately to the left of the last digit moved. The asterisk thus indicates the length of the field addressed, provided it is less than 14 digits long.

D. Minitrace 3 sets word marks in 001 and 201

Word marks are set in core location 001 and 201 and remain set during execution of the program being traced. The latter must therefore be able to function under these conditions.

E. Stacker and printer skip instructions

A Skip After Print instruction is executed directly after the trace of that instruction is printed rather than after the next Print instruction in the program being traced. An Immediate Skip instruction is executed directly before the trace of that instruction is printed.

A Select Stacker instruction given after a Read instruction will not be effective because it will not be executed within the necessary time limit. All cards read should be expected to fall into the normal read pocket.

F. A- and B-operand of 000

Minitrace 3 does not print the contents of location 000 if this position is addressed by the program being traced.

An instruction with this operand (e.g., N 000) will be correctly executed.

G. Word marks must follow all instructions

It is recommended that programs that may be traced be written with word marks in the location following each instruction, thus extending the general requirement to the three instructions that do not orindarily require such a word mark (the 4-digit unconditional Branch, the 7-digit Set Word Mark, and the 7-digit Clear Storage and Branch).

However, Minitrace 3 will usually be able to trace the above instructions if a word mark occurs in core within 32 digits after the last digit of the instruction (within 35 digits of the 4-digit unconditional Branch).

The author's experience has shown that this requirement must especially be kept in mind when writing the last instruction of the program (the instruction preceding the END card) and when patching assembled decks (also remember not to put a patch into locations used by Minitrace 3).

H. Assembly procedure for Minitrace 3

When Minitrace 3 is reassembled (recompiled) in order to relocate it in core, the first three cards of the assembled object deck, which are the standard clear storage and loader cards, are removed and discarded before the deck is used. In order that the Minitrace 3 deck may be loaded together with the deck of the program being traced, both must be assembled by the same method. In other words, an uncondensed SPS Minitrace 3 deck is used with an uncondensed SPS program. Similarly, a Minitrace 3 assembled by Autocoder must be used with a program that has been assembled by Autocoder.

I. Suggestions for tracing programs that exceed 2,875 digits

Programs that in their final form will use most of the 4,000 digits of core space, and which therefore cannot be traced in that form, can sometimes be cut down for debugging and tracing purposes by assembling them without constants, storage areas, or routines that are not used by the part of the program that is giving trouble. At other times, a particularly difficult routine can be first written as a separate small program that is debugged separately and then incorporated into the real program as a subroutine. As an example of this procedure, suppose that a programmer is writing a growth rate program that will use all of the core. He feels that it would be very useful to trace the part of the program that finds the logarithms of the input data. He could very easily write a separate program just to find and print the logarithms of the numbers. After debugging this program and using the trace if necessary, he can then use it as a subroutine in his growth rate program and know that this part of the growth rate program will be trouble-free from the beginning. This two-step method often reduces the total time required.

Programmers with access to a 1401 that is larger than the one they are writing for can make particularly effective use of the Minitrace program. If a programmer is writing for a 4K machine but can arrange to debug his program on a larger machine, he can use the Minitrace 2 program that will be located in the part of the core of the larger machine that he will not be using. In this circumstance, he can trace a program that uses the full 4,000 digits of core of the machine that he is writing for. This situation often arises at IBM test centers, where customers who have purchased 4K or 8K machines are usually given 16K machines to use for testing the programs that they are writing prior to the delivery of their machine.

J. Procedure for tracing only part of a program

To save time in tracing, it is sometimes desirable to be able to trace only part of a program after a preceding part of the program has been executed without tracing. (If part of a program is to be traced without executing the preceding parts, simply change the starting address as outlined under "Operating Procedures".) The following procedure can be followed to stop a program in the middle of a run and start tracing at that point:

1. Remove "End" card from program to be traced.

2. Replace the first card of Minitrace 3 with a card which is identical except that the address at which tracing is to begin is punched into columns 24-26 of an uncondensed SPS Minitrace 3 deck or into columns 1-3 of a condensed SPS or Autocoder Minitrace 3 deck.

3. Put the Minitrace 3 deck behind the program to be traced.

4. Replace the last card of Minitrace 3 with the "End" card from the program to be traced.

5. Place data cards, if any, behind the combined decks.

6. Set address stop to the address at which tracing is to begin.

7. Load and run the program to the address stop. The Start button may have to be pressed twice at address stops during loading.

8. Set the I-address register to 2879, the address to the first instruction of Minitrace 3.

9. Press Start to begin tracing.

To stop tracing before the execution of a given instruction, set address stop to the address of the instruction which <u>follows</u> that instruction. A portion or the remainder of the program being traced may then be properly executed by restarting at the address of the given instruction. Tracing may be resumed later in the program by the following procedure:

1. Set address stop to the address of the instruction at which tracing is to be resumed.

2. Press Start and run the program to the address stop.

3. Put the address of the instruction at which tracing is to be resumed into core location 2876-2878.

4. Reset the I-address register to 2879.

5. Press Start to resume tracing.