DECUS

PROGRAM LIBRARY

DECUS NO.	8-57
TITLE	FIXED POINT TRACE NO. 2
AUTHOR	B.J. Biavati
COMPANY	Radiological Research Laboratory Columbia University New York, New York
DATE	January, 1967
SOURCE LANGUAGE	

FIXED POINT TRACE NO. 2

DECUS Program Write-up

DECUS No. 8-57

General: Fixed Point Trace No. 2 (an expansion of Fixed Point Trace No. 1) is a monitor program which executes the user's program one instruction at a time, as each instruction is carried out the contents of the program counter, the instruction, the contents of the accumulator and link and of the effective address are reported by means of the ASR-33.

The symbolic tape provided has a single origin setting instruction (6000). Any four consecutive memory pages can be used, with the obvious exception of page zero, by changing this one instruction.

Usage:

Assuming the user's program to be in memory the trace program is loaded by way of the binary loader. The starting address (6000) is loaded in the program counter and the computer is started. The program then expects input from the ASR-33 keyboard in the following form.

```
Col 1    Col 2    Col 3    Col 4

xxxx     xxxx     xxxx     xxxx
xxxx     xxxx     xxxx     xxxx
------------
------------
------------
xxxx     xxxx     xxxx     xxxx
(0000)
yyyy
```

Column 1 contains the address at which the trace mode should start.

Column 2 contains the address at which the trade mode should stop.

Column 3 indicates when the region defined in columns 1 and 2 shall be traced.

If 0000: trace the first time it is encountered and not again.

0001: trace every time it is encountered.

0002: trace every 2nd time it is encountered.

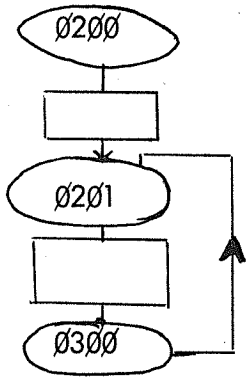
$(n)_8$: trace every (n_{8th}) time.

Column 4 indicates how subroutines will be handled in the defined region.

If not = 0: subroutines are not traced.

If = 0: then subroutine will be traced.

After 15 such entries are made the monitor will expect yyyy which is the starting address in the user's program. If less than 15 entries are required a 0000 in column 1 will indicate to the monitor that the next address entered is yyyy. It is to be recognized that the same starting addresses and stopping address can be entered a number of times, for example, consider the flow chart below:



If we enter

0201	0300	0000	0001
0201	0300	0001	0000
0000			
0200			

then the region from 201 to 300 will be traced the first time through, not tracing all subroutines encountered. The region from 201 to 300 will then be traced each time it is encountered and subroutine tracing will not be suppressed.

Note: The instruction at 0300 will not be traced. In order to trace (300) we would enter:

0201	0201	0001	0000

Restrictions:

1. The program interrupt facility must not be used.
2. In those cases where subroutines are not traced the return must be to the instruction following the JMS instruction or to the following one:

```
JMS       ABLE  }
RETURN          } Legal

JMS       ABLE  }
0               } Legal
RETURN          }

JMS       ABLE  }
0               }
0               } Not legal
RETURN          }
```

3. Any Input-Output transfer by way of the ASR-33 will cause a local disturbance in the trace format.

Output Format:

The following section indicates the output to be expected for each of the five general types of instruction.

Note: The second line is the outputted line.

Symbols:

PC	contents of the program counter
INST	octal instruction
L	contents of the link
ACC.	contents of the accumulator
ADDR	the effective address
C(ADDR)	contents of the effective address

I. AND, TAD, DCA, ISZ

```
PC      INST    L    ACC     C(ADDR)
xxxx    xxxx    x    xxxx    xxxx
```

Note that the instruction is carried out between the type out of INST and L. Therefore the link, accumulator and contents of the effective address are as they are after the completion of the instruction.

II. JMP

```
PC      INST          ADDR
xxxx    xxxx    JMP   xxxx
```

The address is the location to which the jump will transfer.

III. OPR

```
PC      INST          L    ACC
xxxx    xxxx    OPR   x    xxxx
```

Here again the link and accumulator are as they are after the operate instruction is carried out.

IV. JMS

```
PC      INST          ADDR   L    ACC
xxxx    xxxx    JMS   xxxx   x    xxxx
```

In the "subroutine trace mode" the output is as above but L and ACC are not written out. In the "non-trace mode" the subroutine is carried out after the address is printed and then the contents of the link and accumulator are printed. The link and accumulator indicate the situation after the subroutine is completed. Any input or output which might occur in the subroutine will be produced between the writing of the address and link.

V. IOT

```
PC      INST          L    ACC
xxxx    xxxx    IOT   x    xxxx
```

Here again the actual IOT instruction is carried out after printing IOT and before the link is printed. If, for example, there was an output of a carriage return then the link and accumulator contents would overwrite the contents of the program counter and Inst.