ELLIOTT 903

Volume 2: PROGRAMMING INFORMATION

Part 3: UTILITY PROGRAMS

Section 3: MONITOR

~~See NEWSLETTER No 6 : 9~~

See NEWSLETTER No8:13

Contents

Chapter 1: INTRODUCTION

1. 1 Purpose.

MONITOR is a program testing aid which gives the facility of a temporary hold-up at any specified point in the program under test while the contents of the accumulator, Q register, and any specified core locations are output for printing in any combination of integer, octal and instruction modes.

1. 2 Form of Distribution.

The program is distributed as a tape for input by SIR.

1. 3 Method of Use.

A parameter tape is read in by MONITOR to specify the points at which a hold-up is required and the locations and modes then to be output. The program under test is entered and output occurs whenever a monitor point is reached.

An alternative entry point to MONITOR cancels all monitor points previously set.

MONITOR may run at any program level and in any store module. Monitor points must be in the same store module as MONITOR.

1. 4 Entry Points.

0; To read in a parameter tape and set monitor points.

1; To cancel all monitor points and restore the test program to its original form.

Chapter 2: FUNCTIONS

2. 1 Addresses.

All addresses refer to a 64K store and where necessary are taken as modulo 64K. Addresses are always relative to the beginning of the store module in which MONITOR is stored. If MONITOR is in the first module, addresses are absolute. However, if MONITOR is in the second module, then it treats the first module as addresses 56K to 64K, the second module as 0 to 8K, the third module as 8K to 16K, and so on.

2. 2 Parameter List.

A parameter list contains a number of lines which are of four forms.

2. 2. 1 a, b

This form specifies a monitor point.

a is an integer giving the address of a monitor point relative to the beginning of the store module in which MONITOR is stored. $a \leq 8190$. (MONITOR uses a=8191 privately to mark the end of the parameter list.)

b is an integer specifying the output modes. (see 2. 4)

2. 2. 2 a,

This is similar to 2. 2. 1 but b is given the same value as the previous monitor point. b = o if this form is used for the first point.

2. 2. 3 $c_1 \pm c_2 \pm \ldots \pm c_m$

This form specifies the address of one store location whose contents are output whenever its associated monitor point is reached.

If $m = 1$ the address is directly defined.

If $m \geq 2$ the address is indirectly defined as follows:

Take C_1 and treat as an address.

Take the contents of this address, add or subtract C_2 and treat as an address.

Take the contents of this address, add or subtract C_3 and treat as an address.

Take the contents of this address, add or subtract c_n and treat as the defined address.

Each $c \leq 65535$

N. B. This evaluation is done each time that the monitor point is reached during the running of the test program, and may vary from one time to the next.

2. 2. 4 $c_1 \pm c_2 \pm \ldots \pm c_m / d_1 \pm d_2 \pm \ldots \pm d_n$

This form specifies the first and last address of a block of consecutive store addresses whose contents are output whenever its associated monitor point is reached.

Either address can be directly or indirectly defined as in 2. 2. 3. The order is immaterial; MONITOR distinguishes the larger from the smaller.

Each monitor point is specified by forms 2. 2. 1 or 2. 2. 2 and its associated core locations follow with any number (including zero) of forms 2. 2. 3 or 2. 3. 4. Every form must occur on a separate line. Empty lines are ignored. The list ends with a halt character. Up to 10 monitor points may be specified in a parameter list.

2. 3 Character Set for Parameter List.

Meaningful characters are 0 to 9 Ⓛ Ⓗ + - , /

Ⓢ Ⓣ Ⓡ Ⓟ Ⓔ cr are ignored

All other characters are errors.

2. 4 Output

At each monitor point the form of output is determined by the value of the integer b specified in forms 2. 2. 1 or 2. 2. 2. This integer is regarded as a five bit binary number, and the bits have the following significance.

Bit	Decimal Equivalent	Set	Clear
1	1	Layout using spaces.	Layout using tabs.
2	2	Output A and Q registers.	No output of A and Q registers.
3	4	Decimal Output.	No decimal output.
4	8	Octal output.	No Octal output.
5	16	Instruction output	No instruction output.

2. 4. 1 +1

A line of output may contain a store address and the contents of this address in up to three different modes. It is convenient for the user to produce a hard copy aligned in vertical columns. If bit 1=0, this alignment is done by outputting a tab character where appropriate, and a new line is started with a single Ⓛ character. This is the simplest and most convenient output where print-up is done on a Friden Flexowriter.

If bit 1 = 1, the alignment is done by outputting the required number of space characters, and new lines begin with a cr lf blank sequence. This method is required for a Teletype Model 33.

2. 4. 2 +2

If bit 2 = 1, the contents of the accumulator and the Q Register are output in addition to the store locations specified. Labels A and Q are output for identification.

2. 4. 3 +4

If bit 3 = 1, output is in the form of a signed decimal integer with non-significant zeros suppressed. Values lie in the range - 131072 to + 131071.

2. 4. 4 +8

If bit 4 = 1, output is in the form of an unsigned 6 digit octal number without zero suppression.

2. 4. 5 +16

If bit 5=1, output is in instruction form. Bit 18 of the contents of the location is output as /, bits 14-17 as the function number and bits 1-13 as an unsigned decimal address with zero suppression.

The address of a single location and the first address of a block of locations is always output as a label, and thereafter every address which is a multiple of 5 is labelled. Indirectly defined addresses are evaluated at the time of output, according to the method given in 2. 2. 3, but are not distinguished in the actual output from directly defined addresses.

Note that any combination of Integer, octal and instruction modes may be selected.

e. g. b = 22, i. e. 2 + 4 + 16 will give output using tabs of A, Q and specified store locations in decimal and instruction forms.

The output for each monitor point always commences with * a where a is the monitor point address.

It is not necessary to output the contents of any registers or store locations at a monitor point. In this case the output of the monitor point address is an indication that the test program has reached the monitor point. This can be useful for investigating the continuity of a program.

2. 5 Restore.

A parameter list can be cancelled and all monitor points deleted by entering MONITOR at 1;. The test program is restored to its original state. A new parameter list can now be read in and further monitoring performed. Restore must not be used if an error is detected during the input of a parameter tape.

2. 6 Example

2. 6. 1 Parameter tape.

```
 36, 27
 594
 1586/1594

 38, 30
1000+0/1000+18
1594/1586
 Ⓗ
```

2.6.2 Output tape.

(Location 1000 contains the value 54)

```
*36

A       000000  0   0
Q       202345  8   1253

594     000062  0   50

1586    000033  0   27
        000000  0   0
        000146  0   102
        000016  0   14
1590    000017  0   15
        003067  0   1591
        777772 /15  8186
        000001  0   1
        000062  0   50

*38

A                  6          000006       0   6
Q                  66788      202344       8   1252

54                 122877     357775       14  8189
55                 49750      141126       6   598
                   16980      041124       2   596
                   41550      121116       5   590
                   589        001115       0   589
                  -98106      500306      /4   198
60                 590        001116       0   590
                  -16384      740000      /14  0
                   49751      141127       6   599
                   41553      121121       5   593
                   32999      100347       4   231
65                 8792       021130       1   600
                   57413      160105       7   69
                   33359      101117       4   591
                   65607      200107       8   71
                   33359      101117       4   591
70                 16980      041124       2   596
                   596        001124       0   596
                   114690     340002       14  2

1586               30         000036       0   30
                   0          000000       0   0
                   109        000155       0   109
                   14         000016       0   14
1590               15         000017       0   15
                   1591       003067       0   1591
                  -6          777772      /15  8186
                   0          000000       0   0
                   50         000062       0   50
```

Chapter 3: ERRORS

Whenever an error is detected, a message is printed on a new line of the teleprinter in the form ERROR n where n is the error number. If this output is diverted to the punch, it will appear as 1" of run-out followed by cr lf Ⓡ ERROR Ⓢ n. All errors are detected during the reading of the parameter tape and cause reading to stop and MONITOR to end at a dynamic stop. The test program is not affected in any way until the complete parameter tape has been read and accepted as free from error. If an error is detected, a new parameter tape must be prepared and read in from the beginning.

No MONITOR errors can occur while a test program is running and being monitored.

The following errors are detected.

n = 1	odd parity characters
2	Unused characters.
3	Any c or d $\geq$ 65536 in 2. 2. 3 or 2. 2. 4
4	Integer missing
5	Impermissible sequence of separators + - , /
6	b $\geq$ 32 in 2. 2. 1
7	Number of monitor points $\geq$ 11
8	a $\geq$ 8191 in 2. 2. 1 or 2. 2. 2

It is not regarded as an error to specify the same monitor point more than once, although there is little advantage in doing so. MONITOR would be entered separately for each occurrence and the appropriate output generated. However, entry at 1; would not restore the test program to its original form correctly.

Chapter 4: METHOD USED

At entry 0; the parameter tape is read, and analysed and the information stored in a buffer (see 6). If no errors are detected, the instruction in the test program at each monitor point is preserved in the buffer and replaced by a function 8 instruction causing an entry into MONITOR. Each monitor point has a different entry into MONITOR. At entry the A and Q registers and locations 0 to 7 are preserved, the required information is output, A Q and locations 0 to 7 are restored and the true instruction of the test program is obeyed from within MONITOR workspace. If this instruction causes a jump, then the test program is re-entered at the jump point; otherwise the following instruction in MONITOR workspace is a function 8 instruction causing re-entry to the test program at the instruction immediately following the monitor point.

Note that it is inadvisable to monitor a function 11 instruction because the SCR value stored refers to MONITOR workspace and not the true position in the test program.

Monitor points can be on more than one level of program provided they are chosen so that MONITOR is never entered at one level when it has been interrupted in monitoring at a lower level.

For a large program operating at several levels or in several store modules, several independent MONITOR routines may be used to cover different modules or levels. Monitoring at differing levels may lead to a certain amount of jumbling in the output.

At entry 1; all the monitor points in the buffer are scanned and the preserved instructions of the test program restored.

Chapter 5: OPERATING INSTRUCTIONS

Load test program and MONITOR.

Load parameter tape and enter MONITOR at 0; The correction tape is read and checked until Ⓗ is read, when MONITOR ends at a dynamic stop.

Enter test program which will run at normal speed except for hold-up during MONITOR output whenever a monitor point is reached.

Enter MONITOR at 1; to delete all monitor points and restore the test program to its true state.

Chapter 6: STORE USED

The relocatable version uses 612 locations for program. It stores the information on the parameter tape immediately after itself in store. Each monitor point specified by 2. 2. 1 or 2. 2. 2 uses three locations (for the monitor point address, the displaced instruction of the test program, and the output mode respectively). Addresses specified by 2. 2. 3 and 2. 2. 4 take m and m+n locations respectively. One location is used finally to mark the end of the parameter list.

Chapter 7: TIME TAKEN

The parameter tape is read at the speed of the reader. The monitoring output, is punched at the speed of the punch; the test program otherwise runs at normal speed. Restore time is a few millisecs.