


```
0001 SUBROUTINE CLASSIFY (LUN,SBI_REGA)
0002 C
0003 C Version: 'V04-000'
0004 C
0005 C*****
0006 C*
0007 C* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
0008 C* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
0009 C* ALL RIGHTS RESERVED. *
0010 C*
0011 C* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0012 C* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
0013 C* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
0014 C* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0015 C* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
0016 C* TRANSFERRED. *
0017 C*
0018 C* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
0019 C* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
0020 C* CORPORATION. *
0021 C*
0022 C* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
0023 C* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
0024 C*
0025 C*
0026 C*****
0027 C
0028 C
0029 C AUTHOR BRIAN PORTER CREATION DATE 11-DEC-1979
0030 C
0031 C
0032 C++
0033 C Functional description:
0034 C
0035 C This module is called to identify SBI adapters.
0036 C
0037 C Modified by:
0038 C
0039 C v03-002 SAR0067 Sharon A. Reynolds, 20-Jun-1983
0040 C Changed the carriage control in the 'format' statements
0041 C for use with ERF.
0042 C
0043 C v03-001 BP0004 Brian Porter, 19-FEB-1982
0044 C Added ci780 support.
0045 C**
0046 C--
0047 C
0048 C
0049 C
0050 C
0051 C BYTE LUN
0052 C
0053 C INTEGER*4 SBI_REGA
0054 C
0055 C INTEGER*4 FIELD
0056 C
0057 C parameter ci780 = 56
```

CRY
PRC
C
ENT
C
VAR
AP
AP
ARR
AP
AP
FUN
1

```
0058
0059     PARAMETER      DR780 = 48
0060
0061     PARAMETER      MBA = 32
0062
0063     PARAMETER      UBA_0 = 40
0064
0065     PARAMETER      UBA_3 = 43
0066
0067
0068     c
0069     c      Extract adapter type information
0070     c
0071
0072     field = lib$extzv(0,8,sbi_rega)
0073
0074     c
0075     c      Is it the DR780?
0076     c
0077
0078     if (field .eq. dr780) then
0079     call dr780_rega (lun,sbi_rega)
0080
0081
0082     c
0083     c      Is it the RH780?
0084     c
0085
0086     else if (field .eq. mba) then
0087     call rh780_configuration_register (lun,sbi_rega)
0088
0089
0090     c
0091     c      Is it the DW780?
0092     c
0093
0094     else if (field .ge. uba_0 .and. field .le. uba_3) then
0095     call uba_rega (lun,sbi_rega)
0096
0097
0098     c
0099     c      Is it the CI780?
0100     c
0101
0102     else if (field .eq. ci780) then
0103     call ci780_rega (lun,sbi_rega)
0104
0105
0106     c
0107     c      Is it a memory?
0108     c
0109
0110     else if (lib$extzv(5,3,sbi_rega) .eq. 0) then
0111     call ms780c_rega (lun,sbi_rega)
0112
0113
0114     else if (lib$extzv(5,3,sbi_rega) .eq. 2) then
```


CLASSIFY

E 4
 16-Sep-1984 00:18:49
 5-Sep-1984 13:50:15

VAX-11 FORTRAN V3.4-56
 DISK\$VMSMASTER:[ERF.SRC]CLASSIFY.FOR;1

```

0115
0116      call ma780_rega (lun,sbi_rega)
0117
0118      else if (lib$extzv(5,3,sbi_rega) .eq. 3) then
0119
0120      call ms780e_rega (lun,sbi_rega)
0121
0122      c
0123      c      The if-then-else should be expanded at this point to add
0124      c      new adapter support.
0125      c
0126
0127      ELSE
0128
0129      CALL LINCHK (LUN,2)
0130
0131      WRITE(LUN,70) SBI_REGA
0132      70  FORMAT(/' ',T8,'SBI REG 'A'',T24,Z8.8)
0133      ENDIF
0134
0135      RETURN
0136
0137      END
    
```

PROGRAM SECTIONS

Name	Bytes	Attributes
0 \$CODE	298	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 \$PDATA	45	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 \$LOCAL	60	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
Total Space Allocated	403	

ENTRY POINTS

Address	Type	Name
0-00000000		CLASSIFY

VARIABLES

Address	Type	Name	Address	Type	Name	Address	Type	Name
2-00000000	I*4	FIELD	AP-00000004@	L*1	LUN	AP-00000008@	I*4	SBI_REGA

CLASSIFY

F 4
16-Sep-1984 00:18:49
5-Sep-1984 13:50:15

VAX-11 FORTRAN V3.4-56
DISK\$VMSMASTER:[ERF.SRC]CLASSIFY.FOR;1

Page 4

STR
V04

LABELS

Address	Label
1-00000014	70'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name
	CI780_REGA		DR780_REGA	I*4	LIB\$EXTZV
	LINCHR		MA780_REGA		MS780C_REGA
	MS780E_REGA		RH780_CONFIGURATION_REGISTER		UBA_REGA

COMMAND QUALIFIERS

FORTRAN /LIS=LIS\$:CLASSIFY/OBJ=OBJ\$:CLASSIFY MSRC\$:CLASSIFY

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
 /DEBUG=(NOSYMBOLS,TRACEBACK)
 /STANDARD=(NOSYNTAX,NOSOURCE FORM)
 /SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP)
 /F77 /NOG_FLOATING /I4 /OPTIMIZE /WARNINGS /NOD_LINES /NOCROSS_REFERENCE /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 1.39 seconds
 Elapsed Time: 4.93 seconds
 Page Faults: 94
 Dynamic Memory: 169 pages

A dense grid of 16 columns and 16 rows of small, faded terminal window screenshots. Each window displays a different program or utility interface, including:

- CLASSIFY LIS**: A window with various data fields and text.
- DR750 LIS**: A window with a large block of text.
- DR780 LIS**: A window with a large block of text.
- DR11W LIS**: A window with a large block of text.
- DTAILS LIS**: A window with a large block of text.
- DUMPREG LIS**: A window with a large block of text.
- DUTDIRTUR LIS**: A window with a large block of text.
- DUP3271 LIS**: A window with a large block of text.
- DUP11 LIS**: A window with a large block of text.
- CRYPTK LIS**: A window with a large block of text.
- DODISKS LIS**: A window with a large block of text.
- CSTRING LIS**: A window with a large block of text.
- DHEADS LIS**: A window with a large block of text.
- COMPRESS LIS**: A window with a large block of text.
- DECODECC LIS**: A window with a large block of text.
- CALCMAP LIS**: A window with a large block of text.

Each window also contains various status indicators, error messages, and data lists, though they are mostly illegible due to fading.