```
EEEEEEEEEEEEEEE  RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEEEEEEEEEEEEEE  RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEEEEEEEEEEEEEE  RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEEEEEEEEEEE     RRRRRRRRRRRR     FFFFFFFFFFFF
EEEEEEEEEEEE     RRRRRRRRRRRR     FFFFFFFFFFFF
EEEEEEEEEEEE     RRRRRRRRRRRR     FFFFFFFFFFFF
EEE              RRR   RRR        FFF
EEE              RRR   RRR        FFF
EEE              RRR   RRR        FFF
EEE              RRR      RRR     FFF
EEE              RRR      RRR     FFF
EEE              RRR      RRR     FFF
EEEEEEEEEEEEEEE  RRR         RRR  FFF
EEEEEEEEEEEEEEE  RRR         RRR  FFF
EEEEEEEEEEEEEEE  RRR         RRR  FFF
```

```
MM      MM  BBBBBBBB      AAAAAA      IIIIII    NN      NN  TTTTTTTTTT
MM      MM  BBBBBBBB      AAAAAA      IIIIII    NN      NN  TTTTTTTTTT
MMMM  MMMM  BB      BB  AA      AA      II      NN      NN      TT
MMMM  MMMM  BB      BB  AA      AA      II      NN      NN      TT
MM  MM  MM  BB      BB  AA      AA      II      NNNN    NN      TT
MM  MM  MM  BB      BB  AA      AA      II      NNNN    NN      TT
MM      MM  BBBBBBBB    AA      AA      II      NN  NN  NN      TT
MM      MM  BBBBBBBB    AA      AA      II      NN  NN  NN      TT
MM      MM  BB      BB  AAAAAAAAAA      II      NN    NNNN      TT
MM      MM  BB      BB  AAAAAAAAAA      II      NN    NNNN      TT
MM      MM  BB      BB  AA      AA      II      NN      NN      TT          ....
MM      MM  BB      BB  AA      AA      II      NN      NN      TT          ....
MM      MM  BBBBBBBB    AA      AA    IIIIII    NN      NN      TT          ....
MM      MM  BBBBBBBB    AA      AA    IIIIII    NN      NN      TT          ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0001    C
0002    C Version:      'V04-000'
0003    C
0004    C*****************************************************************************
0005    C*                                                                           *
0006    C*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0007    C*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0008    C*  ALL RIGHTS RESERVED.                                                     *
0009    C*                                                                           *
0010    C*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0011    C*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0012    C*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0013    C*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0014    C*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0015    C*  TRANSFERRED.                                                             *
0016    C*                                                                           *
0017    C*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0018    C*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0019    C*  CORPORATION.                                                             *
0020    C*                                                                           *
0021    C*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0022    C*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0023    C*                                                                           *
0024    C*                                                                           *
0025    C*****************************************************************************
0026    C
0027
0028    c       Author  Brian Porter                    Creation date   18-NOV-1981
0029
0030    c++
0031    c       Functional description:
0032    c
0033    c       This module displays massbus adapter interrupts which occur
0034    c       on behalf of itself.
0035    c
0036    c       Modified by:
0037    C
0038    C       V03-003 SAR0147         Sharon A. Reynolds,     5-Oct-1983
0039    C               Added an SYE update that makes the register heralds
0040    C               generic and adds support for 11/7xx.
0041    C
0042    C       V03-002 SAR0081         Sharon A. Reynolds,     20-Jun-1983
0043    C               Changed the carriage control in the 'format' statements
0044    C               for use with ERF.
0045    C
0046    C       V03-001 SAR0055         Sharon A. Reynolds,     13-Jun-1983
0047    C               Removed brief/cryptic support.
0048    C
0049    c--
0050    c--
0051
0052
0053
0054            Subroutine MBA_INTERRUPTS (lun,option)
0055
0056
0057            include 'src$:msghdr.for /nolist'
```

```
0116
0117
0118            byte            lun
0119
0120            character*1     option
0121
0122
0123            if (
0124            1 lib$extzv(24,8,emb$l_hd_sid) .eq. 255
0125            1 .or.
0126            1 lib$extzv(24,8,emb$l_hd_sid) .eq. 1
0127            1 .or.
0128            1 LIB$EXTZV(24,8,emb$l_hd_sid) .EQ. 4
0129            1 ) then
0130
0131            if (option .eq. 'S') then
0132
0133            call rh780_interrupt (lun)
0134            endif
0135    c
0136    c       For future expansion the IF-THEN-ELSE should be
0137    c       extended here.
0138    c
0139
0140            endif
0141
0142            return
0143
0144            end
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	80	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	8	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	32	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3	EMB	512	PIC OVR REL GBL SHR NOEXE RD WRT LONG
	Total Space Allocated	632	

ENTRY POINTS

Address	Type	Name
0-00000000		MBA_INTERRUPTS

VARIABLES

Address	Type	Name	Address	Type	Name
3-00000000	I*4	EMB$L_HD_SID	3-00000004	I*2	EMB$W_HD_ENTRY
3-0000000E	I*2	EMB$W_HD_ERRSEQ	AP-0000004a	L*1	LUN
AP-00000008a	CHAR	OPTION			

ARRAYS

Address	Type	Name	Bytes	Dimensions
3-00000000	L*1	EMB	512	(0:511)
3-00000006	I*4	EMB$Q_HD_TIME	8	(2)

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name
I*4	LIB$EXTZV		RH780_INTERRUPT

```
0001
0002
0003
0004    c++
0005    c       This routine displays the massbus interrupt error log entry for the
0006    c       11/780.  The format of the error log record is as
0007    c       follows.
0008    c
0009    c       +--------------------------------+
0010    c       |      configuration register    |
0011    c       +--------------------------------+
0012    c       |         control register       |
0013    c       +--------------------------------+
0014    c       |          status register       |
0015    c       +--------------------------------+
0016    c       |     virtual address register   |
0017    c       +--------------------------------+
0018    c       |       byte count register      |
0019    c       +--------------------------------+
0020    c       |      diagnostic register       |
0021    c       +--------------------------------+
0022    c       |      selected map register     |
0023    c       +--------------------------------+
0024    c       |    command address register    |
0025    c       +--------------------------------+
0026    c--
0027
0028
0029
0030
0031            Subroutine RH780_INTERRUPT (lun)
0032
0033
0034            include 'src$:msghdr.for /nolist'
0093
0094
0095            byte            lun
0096
0097            integer*4       rh780_control_register(6)
0098            integer*4       diagnostic_register
0099            integer*4       interrupt_pc
0100            integer*4       interrupt_psl
0101            integer*4       adapter_tr
0102            integer*4       dummy_variable
0103            integer*4       compress4
0104
0105            logical*1       diagnostic_mode
0106
0107            equivalence     (emb(16),rh780_control_register)
0108            equivalence     (emb(36),diagnostic_register)
0109            equivalence     (emb(40),adapter_tr)
0110            equivalence     (emb(44),interrupt_pc)
0111            equivalence     (emb(48),interrupt_psl)
0112
0113
0114            call frctof (lun)
0115
```

```
0116            call header (lun)
0117
0118            call logger (lun,''RH'' INTERRUPT')
0119
0120            diagnostic_mode = .false.
0121
0122            if (lib$extzv(3,1,rh780_control_register(2)) .eq. 1)
0123            1 diagnostic_mode = .true.
0124
0125            if (lib$extzv(0,4,diagnostic_register) .ne. 0) diagnostic_mode = .true.
0126
0127            call linchk (lun,2)
0128
0129            write(lun,5) interrupt_pc
0130    5       format(/' ',t8,'ERROR PC',t24,z8.8)
0131
0132            if (.not. diagnostic_mode) then
0133
0134            call vaxpsl (lun,interrupt_psl)
0135
0136            call linchk (lun,2)
0137
0138            write(lun,10) adapter_tr
0139    10      format(/' ',t8,'ADAPTER TR# ',i<compress4 (adapter_tr)>,'.')
0140
0141            call rh780_control_registers (lun,rh780_control_register,
0142            1 dummy_variable)
0143            else
0144
0145            call linchk (lun,11)
0146
0147            write(lun,15) interrupt_psl,adapter_tr,
0148            1 (rh780_control_register(i),i = 1,2)
0149    15      format(/' ',t8,'ERROR PSL',t24,z8.8,//,
0150            1 t8,'ADAPTER TR# ',i<compress4 (adapter_tr)>,'.',//,
0151            1 t8,''RH'' CSR',t24,z8.8,/,
0152            1 t8,''RH'' CR',t24,z8.8)
0153
0154            if (lib$extzv(3,1,rh780_control_register(2)) .eq. 1) then
0155
0156            call linchk (lun,1)
0157
0158            write(lun,30)
0159            endif
0160
0161            write(lun,20) (rh780_control_register(i),i = 3,5)
0162    20      format(' ',t8,''RH'' SR',t24,z8.8,/,
0163            1 t8,''RH'' VAR',t24,z8.8,/,
0164            * t8,''RH'' BCR',t24,z8.8)
0165            endif
0166
0167            call linchk (lun,1)
0168
0169            write(lun,25) diagnostic_register
0170    25      format(' ',t8,''RH'' DR',t24,z8.8)
0171
0172            if (lib$extzv(0,4,diagnostic_register) .ne. 0) then
```

```
0173
0174            call linchk (lun,1)
0175
0176            write(lun,30)
0177     30     format(' ',t40,'DIAGNOSTIC MODE')
0178            endif
0179
0180            return
0181            end
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	518	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	276	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	144	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3	EMB	512	PIC OVR REL GBL SHR NOEXE RD WRT LONG
	Total Space Allocated	1450	

ENTRY POINTS

Address	Type	Name
0-00000000		RH780_INTERRUPT

VARIABLES

Address	Type	Name	Address	Type	Name
3-00000028	I*4	ADAPTER_TR	2-00000000	L*1	DIAGNOSTIC_MODE
3-00000024	I*4	DIAGNOSTIC_REGISTER	2-00000004	I*4	DUMMY_VARIABLE
3-00000000	I*4	EMB$L_HD_SID	3-00000004	I*2	EMB$W_HD_ENTRY
3-0000000E	I*2	EMB$W_HD_ERRSEQ	2-00000008	I*4	I
3-0000002C	I*4	INTERRUPT_PC	3-00000030	I*4	INTERRUPT_PSL
AP-0000004@	L*1	LUN			

ARRAYS

Address	Type	Name	Bytes	Dimensions
3-00000000	L*1	EMB	512	(0:511)
3-00000006	I*4	EMB$Q_HD_TIME	8	(2)
3-00000010	I*4	RH780_CONTROL_REGISTER	24	(6)

LABELS

Address	Label	Address	Label	Address	Label	Address	Label	Address	Label	Address	Label
1-00000027	5'	1-0000003D	10'	1-0000005B	15'	1-000000B1	20'	1-000000E9	25'	1-000000FD	30'

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name
I*4	COMPRESS4		FRCTOF		HEADER
I*4	LIB$EXTZV		LINCHK		LOGGER
	RH780_CONTROL_REGISTERS		VAXPSL		

COMMAND QUALIFIERS

```
  FORTRAN /LIS=LIS$:MBAINT/OBJ=OBJ$:MBAINT MSRC$:MBAINT

  /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:              2.75 seconds
  Elapsed Time:          8.40 seconds
  Page Faults:           144
  Dynamic Memory:        179 pages
```

0150 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

LPA11K
LIS

MCHECKS
LIS

LCPRINTER
LIS

MBA
LIS

LOGMSCP
LIS

LINCHK
LIS

MBAINT
LIS

LABEL
LIS

LOGGER
LIS

MASSTAPE
LIS

MASSDISK
LIS