F 15

```
HHH         HHH  LLL              DDDDDDDDDDDD
HHH         HHH  LLL              DDDDDDDDDDDD
HHH         HHH  LLL              DDDDDDDDDDDD
HHH         HHH  LLL              DDD         DDD
HHH         HHH  LLL              DDD         DDD
HHH         HHH  LLL              DDD         DDD
HHH         HHH  LLL              DDD         DDD
HHH         HHH  LLL              DDD         DDD
HHH         HHH  LLL              DDD         DDD
HHHHHHHHHHHHHHH  LLL              DDD         DDD
HHHHHHHHHHHHHHH  LLL              DDD         DDD
HHHHHHHHHHHHHHH  LLL              DDD         DDD
HHH         HHH  LLL              DDD         DDD
HHH         HHH  LLL              DDD         DDD
HHH         HHH  LLL              DDD         DDD
HHH         HHH  LLL              DDD         DDD
HHH         HHH  LLL              DDD         DDD
HHH         HHH  LLL              DDD         DDD
HHH         HHH  LLLLLLLLLLLLLLL  DDDDDDDDDDDD
HHH         HHH  LLLLLLLLLLLLLLL  DDDDDDDDDDDD
HHH         HHH  LLLLLLLLLLLLLLL  DDDDDDDDDDDD
```

```
**FILE**ID**HLDMAIN                                  G 3





HH      HH  LL          DDDDDDDD    MM      MM    AAAAAA      IIIIII    NN      NN
HH      HH  LL          DDDDDDDD    MM      MM    AAAAAA      IIIIII    NN      NN
HH      HH  LL          DD      DD  MMMM  MMMM  AA      AA      II      NN      NN
HH      HH  LL          DD      DD  MMMM  MMMM  AA      AA      II      NN      NN
HH      HH  LL          DD      DD  MM  MM  MM  AA      AA      II      NNNN    NN
HH      HH  LL          DD      DD  MM  MM  MM  AA      AA      II      NNNN    NN
HHHHHHHHHH  LL          DD      DD  MM      MM  AA      AA      II      NN  NN  NN
HHHHHHHHHH  LL          DD      DD  MM      MM  AA      AA      II      NN  NN  NN
HH      HH  LL          DD      DD  MM      MM  AAAAAAAAAA      II      NN    NNNN
HH      HH  LL          DD      DD  MM      MM  AAAAAAAAAA      II      NN    NNNN
HH      HH  LL          DD      DD  MM      MM  AA      AA      II      NN      NN      ....
HH      HH  LL          DD      DD  MM      MM  AA      AA      II      NN      NN      ....
HH      HH  LLLLLLLLLL  DDDDDDDD    MM      MM  AA      AA    IIIIII    NN      NN      ....
HH      HH  LLLLLLLLLL  DDDDDDDD    MM      MM  AA      AA    IIIIII    NN      NN      ....

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
```

Table of contents

```
0000            .TITLE  HLDMAIN - HLD MAINLINE
0000            .IDENT  'V04-000'
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000
0000    ;++
0000    ; FACILITY: DECNET HOST LOADER (HLD)
0000    ;
0000    ; ABSTRACT:
0000    ;
0000    ;       HLD IS A COMPONENT OF DECNET/VAX-11. IT PROVIDES ACCESS TO
0000    ;       RSX11S TASK IMAGES STORED ON A VAX SYSTEM.
0000    ;
0000    ; ENVIRONMENT:
0000    ;
0000    ;       THE HLD IMAGE EXECUTES IN THE CONTEXT OF A PROCESS CREATED BY
0000    ;       NETACP. IT RUNS IN USER MODE AND REQUIRES NETWORK PRIVILEGE.
0000    ;
0000    ; AUTHOR: SCOTT G. DAVIS,       CREATION DATE: 10-MAY-79
0000    ;
0000    ; MODIFICATIONS:
0000    ;
0000    ;       V001    SGD0003         19-Mar-1984
0000    ;               Cleanup log on normal termination
0000    ;--
0000    ;
0000    ; INCLUDE FILES
0000    ;
0000    ;
0000    ; MACROS:
0000    ;
0000    ;       NONE
```

```
                        0000         .SBTTL  HLD$START - MAINLINE
                    00000000         .PSECT  HLD$CODE        NOSHR,EXE,RD,NOWRT
                        0000
                        0000   ;++
                        0000   ; FUNCTIONAL DESCRIPTION:
                        0000   ;
                        0000   ;       HLD$START IS RESPONSIBLE FOR HLD INITIALIZATION AND TERMINATION.
                        0000   ;
                        0000   ; CALLING SEQUENCE:
                        0000   ;
                        0000   ;       HLD IS INITIATED BY NETACP.
                        0000   ;
                        0000   ; INPUT PARAMETERS:
                        0000   ;
                        0000   ;       NONE
                        0000   ;
                        0000   ; IMPLICIT INPUTS:
                        0000   ;
                        0000   ;       LOGICAL NAME 'SYS$NET'
                        0000   ;
                        0000   ;--
                        0000
                  0000  0000         .ENTRY  HLD$START,^M<>          ; ENTRY POINT FROM EXECUTIVE
                        0002
                        0002   ;
                        0002   ; OPEN THE PRINT FILE
                        0002   ;
                        0002
        50   0000'CF  9E  0002         MOVAB   W^HLD$PRTFAB,R0         ; SET TO OPEN FILE
                FFF6'  30  0007         BSBW    HLD$RMS_OPEN            ; DO IT
        50   0000'CF  9E  000A         MOVAB   W^HLD$PRTRAB,R0         ; SET TO CONNECT REACORD STREAM
                FFEE'  30  000F         BSBW    HLD$RMS_CONNECT         ; DO IT
                        0012   ;
                        0012   ; CREATE A CONTROL/INFORMATION PATH TO NETACP IN PREPARATION FOR
                        0012   ; NON-TRANSPARENT NETWORK I/O. ALSO ASSOCIATE A MAILBOX WITH THE
                        0012   ; CHANNEL.
                        0012   ;
                        0012
                        0012         $ASSIGN_S DEVNAM=W^HLD$GQ_LNKNAM- ; "NET0:" REFERS TO NETACP
                        0012                 CHAN=W^HLD$GW_LNKCHN    ; STORE CHANNEL #
                        0023         CHECK_SS                        ; CHECK STATUS CODE
                        0026
                        0026   ;
                        0026   ; TRANSLATE THE LOGICAL NAME 'SYS$NET'. ITS EQUIVALENCE STRING IS
                        0026   ; SET-UP BY NETACP TO BE THE ENTIRE NETWORK CONNECT BLOCK (NCB) WHICH
                        0026   ; IS REQUIRED TO GAIN NON-TRANSPARENT ACCESS TO THE NETWORK.
                        0026   ;
                        0026
        57   0000'CF  7E  0026         MOVAQ   W^HLD$GQ_NCBDESC,R7     ; GET ADDRESS OF NCB DESCRIPTOR
           67    40 8F  9A  002B         MOVZBL  #64,(R7)                ; STORE BUFFER SIZE
     04 A7   0000'CF  9E  002F         MOVAB   W^HLD$GT_NCBBUF,4(R7)   ; STORE BUFFER ADDRESS
                        0035         $TRNLOG_S LOGNAM=W^HLD$GQ_SYSNAM- ; ADDRESS OF LOGICAL NAME DESCRIPTOR
                        0035                 RSLLEN=(R7)-            ; UPDATE SIZE IN NCB DESCRIPTOR
                        0035                 RSLBUF=(R7)             ; PUT EQUIVALENCE STRING IN NCB
                        004A         CHECK_SS                        ; CHECK STATUS CODE
                        004D
                        004D   ;++
                          004D  ; THE GENERAL FORMAT OF THE EQUIVALENCE STRING DERIVED FROM SYS$NET IS:
                          004D  ;
                          004D  ;       nodespec::"objecttype=taskid/netacp_string"
                          004D  ;       userdata (before close ") consists of:
                          004D  ;               4 bytes of rad50 task name
                          004D  ;               1 word of partition address (0=>mapped)
                          004D  ;               1 word of partition size (in bytes or 32.-word blocks)
                          004D  ;               1 byte of lun (general purpose) flag - 0=no,1=>lur fixing
                          004D  ;               1 byte of function code (-1=overlay,0=read, 1=chkread, 2=chkwrt)
                          004D  ;               2 words of VBN if overlay request
                          004D
                          004D  ; On return, the optional data consists of:
                          004D  ;               1 word of transfer size
                          004D  ;               1 byte of no. of luns to fix
                          004D  ;
                          004D  ; THE FOLLOWING WILL BE OUTPUT TO THE PRINT FILE:
                          004D  ;
                          004D  ;       nodespec::"objecttype=taskid"
                          004D  ;
                          004D  ; THE FOLLOWING WILL BE PUT INTO THE NCB FOR A CONNECT ACCEPT
                          004D  ;
                          004D  ;       nodespec::"objecttype=taskid/next_two_bytes"
                          004D  ; userdata consists of actual transfer size in place of partition size
                          004D  ;--
                          004D
   04 B7   67    3A   3A  004D          LOCC    #^A\:\,(R7),@4(R7)      ; FIND NODE DELIMITER
                 11   13  0052          BEQL    20$                     ; IF EQL NOT FOUND
0000'CF    67    50   A3  0054          SUBW3   R0,(R7),W^HLD$GQ_NODEDESC ; STORE NAME LENGTH
     0000'CF     57   D0  005A          MOVL    R7,W^HLD$GL_IOPARAM2    ; ADDRESS OF NCB DESCRIPTOR
     61    50    2F   3A  005F          LOCC    #^A\/\,R0,(R1)          ; FIND TASKID DELIMITER
                 07   12  0063          BNEQ    30$                     ; BRANCH IF SLASH FOUND
                          0065  20$:
     50   0000'8F     3C  0065          MOVZWL  #SS$_IVDEVNAM,R0        , MAKE THIS THE EXIT ERROR STATUS
                 64   11  006A          BRB     HLD$EXIT_TO_VMS         ; FATAL ERROR!!
                          006C  ;
                          006C  ; RESPOND TO THE CONNECT INITIATE WITH A CONNECT CONFIRM (WITHOUT USERDATA)
                          006C  ; TO COMPLETE THE LOGICAL LINK.
                          006C  ;
                          006C
                          006C  30$:
           51    03   C0  006C          ADDL    #3,R1                   ; SKIP "/" AND 2 RESERVED BYTES
                          006F  ;
                          006F  ; NOW VERIFY THE CONNECT AND CHECK THE DATA FILE
                          006F  ;
               FF8E'  30  006F          BSBW    HLD$GET_FILE            ; SCAN HLD.DAT
               FF8B'  30  0072          BSBW    HLD$OPEN_TSKFIL         ; OPEN AND PROCESS THE TASK FILE
                          0075  ;
                          0075  ;       CONFIRM THE LOGICAL LINK
                          0075  ;
               FF88'  30  0075          BSBW    HLD$NET_IO              ; CONFIRM THE LINK
                          0078  ;
                          0078  ; Now fix it so that HLD waits for disconnect if transfer is to host.
                          0078  ;
                          0078          $SETEF_S        EFN=#2          ; Set event flag as default
     00'  0000'CF     B1  0081          CMPW    W^HLD$GW_SAVEFUNC,S^#IO$_READVBLK ; Am I reading?
                 1E   12  0086          BNEQ    40$                     ; If NEQ no
                          0088  ;
```

```
                              0088   ; Issue a phony QIO which will terminate when the link is broken
                              0088   ;
                              0088           $QIO_S  -                               ; ISSUE THE REQUEST
                              0088                   CHAN=   W^HLD$GW_LNKCHN- ;
                              0088                   FUNC=   S^#IO$_WRITEVBLK -; Do a write
                              0088                   EFN=    #2-              ;
                              0088                   P1=     .-
                              0088                   P2=     #0
                              00A6   40$:
                              00A6   ;
                              00A6   ; NOW DO THE ACTUAL TASK TRANSFER
                              00A6   ;
0000'CF    0000'CF       9E   00A6           MOVAB   W^HLD$AB_BUFFER,W^HLD$GL_IOPARAM1 ; SET BUFFER ADDRESS
0000'CF    0000'CF       B0   00AD           MOVW    W^HLD$GW_SAVEFUNC,W^HLD$GW_IOFUNC ; SET UP TRANSFER FUNCTION
                28       10   00B4           BSBB    DO_IO                   ; DO THE TRANSFER
          58    56       7D   00B6           MOVQ    R6,R8                   ; SET UP FOR LUN BLOCK TRANSFER
     0000'CF    02       D0   00B9           MOVL    #2,W^HLD$GL_TSKBKT      ; Set up VBN of LUN block(s)
                1E       10   00BE           BSBB    DO_IO                   ; SEND THEM, MAYBE
                              00C0   ;
                              00C0   ; Now wait, if necessary
                              00C0   ;
                              00C0           $WAITFR_S       EFN=#2
 50    00030001 8F       D0   00C9           MOVL    #^X30001,R0             ; Force ''NORMAL'' to the log file
                              00D0
                              00D0   HLD$EXIT_TO_VMS::                       ;
                50       DD   00D0           PUSHL   R0                      ; SAVE EXIT STATUS CODE
             FF2B'       30   00D2           BSBW    HLD$PRINT               ; PRINT RESULTS
                              00D5           $EXIT_S (SP)+                   ; EXIT TO VMS
```

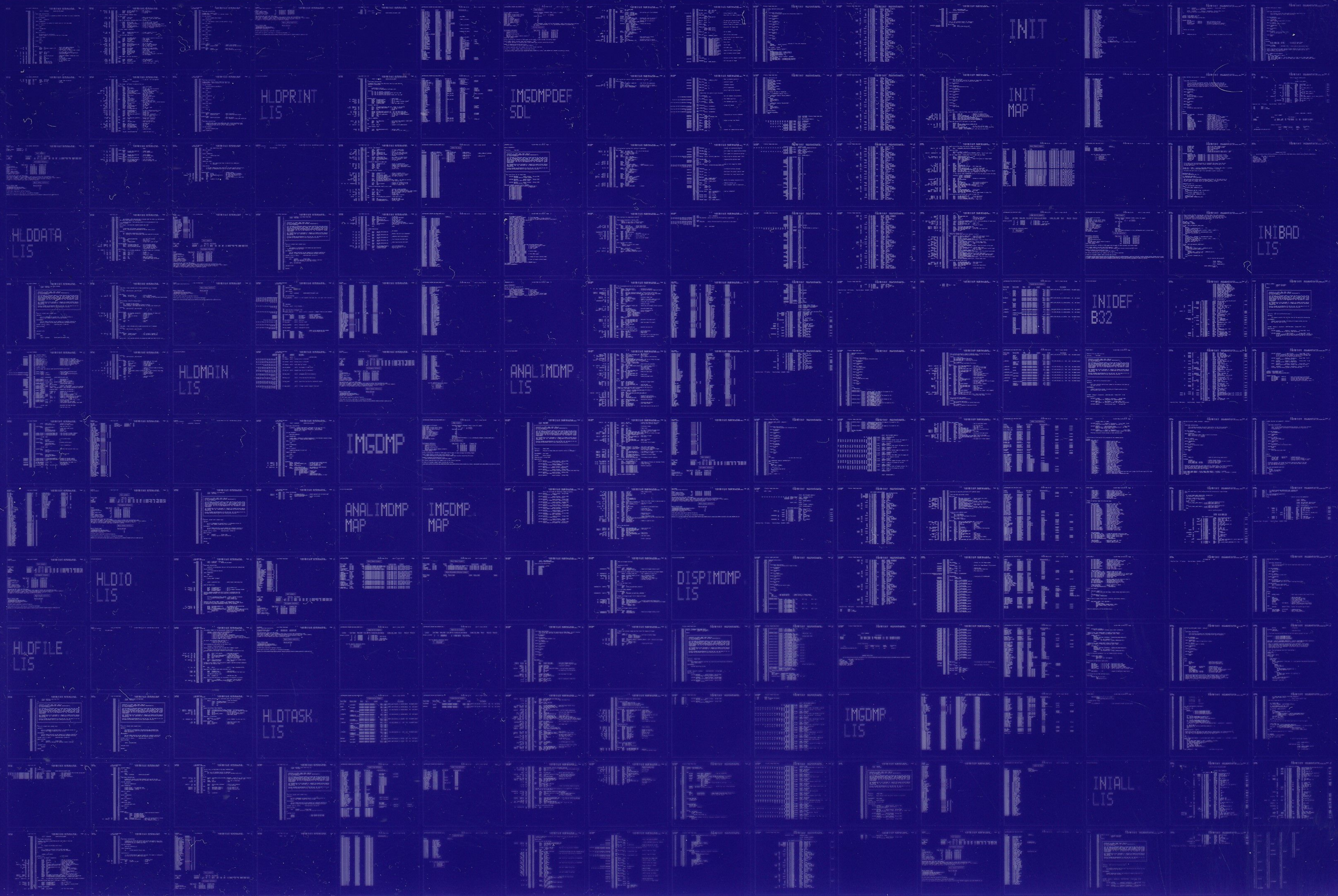
```
                                  00DE   200 ;++
                                  00DE   201 ;
                                  00DE   202 ; DO_IO - CALL I/O ROUTINES TO DO REAL TRANSFER WORK
                                  00DE   203 ;
                                  00DE   204 ; INPUTS.
                                  00DE   205 ;         R8 - NO. OF INTEGRAL BLOCKS TO TRANSFER
                                  00DE   206 ;         R9 - SIZE IN BYTES OF BLOCK AFTER LAST INTEGRAL BLOCK OR 0
                                  00DE   207 ;
                                  00DE   208 ;--
                                  00DE   209
                                  00DE   210 DO_IO:
0000'CF     0000'CF      3C       00DE   211         MOVZWL  W^HLD$GW_IOLEN,W^HLD$GL_IOPARAM2 ; LENGTH OF ALL BUT LAST TRANSFER
                 02      11       00E5   212         BRB     50$                          ; CHECK THE ''NO FULL BLOCK'' CASE
                                  00E7   213 40$:
                 0A      10       00E7   214         BSBB    55$                          ; CALL THE I/O ROUTINES
              FB 58      F4       00E9   215 50$:    SOBGEQ  R8,40$                       ; LOOP
                                  00EC   216
                                  00EC   217
       0000'CF   59      B0       00EC   218         MOVW    R9,W^HLD$GL_IOPARAM2         ; SIZE OF LAST TRANSFER
                 0C      13       00F1   219         BEQL    60$                          ; IF EQL NOTHING
                                  00F3   220 55$:
            0000'DF      16       00F3   221         JSB     @W^HLD$GL_IOROUT_1           ; DO 1ST KIND OF I/O
            0000'DF      16       00F7   222         JSB     @W^HLD$GL_IOROUT_2           ; DO 2ND KIND OF I/O
            0000'CF      D6       00FB   223         INCL    W^HLD$GL_TSKBKT              ; ADVANCE THE BLOCK NO.
                                  00FF   224 60$:
                         05       00FF   225         RSB                                  ; DONE
                                  0100   226
                                  0100   227         .END    HLD$START
```

```
$$T1             = 00000001
DO_IO              000000DE R      01
HLD$AB_BUFFER      ********   X    01
HLD$CHECK_SS       ********   X    01
HLD$EXIT_TO_VMS    000000D0 RG     01
HLD$GET_FILE       ********   X    01
HLD$GL_IOPARAM1    ********   X    01
HLD$GL_IOPARAM2    ********   X    01
HLD$GL_IOROUT_1    ********   X    01
HLD$GL_IOROUT_2    ********   X    01
HLD$GL_TSKBKT      ********   X    01
HLD$GQ_LNKNAM      ********   X    01
HLD$GQ_NCBDESC     ********   X    01
HLD$GQ_NODEDESC    ********   X    01
HLD$GQ_SYSNAM      ********   X    01
HLD$GT_NCBBUF      ********   X    01
HLD$GW_IOFUNC      ********   X    01
HLD$GW_IOLEN       ********   X    01
HLD$GW_LNKCHN      ********   X    01
HLD$GW_SAVEFUNC    ********   X    01
HLD$NET_IO         ********   X    01
HLD$OPEN_TSKFIL    ********   X    01
HLD$PRINT          ********   X    01
HLD$PRTFAB         ********   X    01
HLD$PRTRAB         ********   X    01
HLD$RMS_CONNECT    ********   X    01
HLD$RMS_OPEN       ********   X    01
HLD$START          00000000 RG     01
IO$_READVBLK       ********   X    01
IO$_WRITEVBLK      ********   X    01
SS$_IVDEVNAM       ********   X    01
SYS$ASSIGN         ********  GX    01
SYS$EXIT           ********  GX    01
SYS$QIO            ********  GX    01
SYS$SETEF          ********  GX    01
SYS$TRNLOG         ********  GX    01
SYS$WAITFR         ********  GX    01

                                            +-----------------+
                                            ! Psect synopsis !
                                            +-----------------+

PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
.  ABS  .                       00000000  (    0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
HLD$CODE                        00000100  (  256.)  01 (  1.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD  NOWRT NOVEC BYTE

                                         +------------------------+
                                         ! Performance indicators !
                                         +------------------------+

Phase                     Page faults    CPU Time       Elapsed Time
-----                     -----------    --------       ------------
Initialization                     37    00:00:00.07    00:00:00.95
Command processing                132    00:00:00.59    00:00:04.94
Pass 1                            128    00:00:01.37    00:00:09.38
Symbol table sort                   0    00:00:00.02    00:00:00.01
```

```
Pass 2                            58      00:00:00.58     00:00:02.22
Symbol table output                3      00:00:00.04     00:00:00.04
Psect synopsis output              1      00:00:00.01     00:00:00.01
Cross-reference output             0      00:00:00.00     00:00:00.00
Assembler run totals             361      00:00:02.70     00:00:17.57

The working set limit was 1050 pages.
4983 bytes (10 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 37 non-local and 7 local symbols.
227 source lines were read in Pass 1, producing 16 object records in Pass 2.
11 pages of virtual memory were used to define 11 macros.

                                              +--------------------------+
                                              ! Macro library statistics !
                                              +--------------------------+

Macro library name                             Macros defined
------------------                             --------------
_$255$DUA28:[HLD.OBJ]HLD.MLB;1                            1
_$255$DUA28:[SYSLIB]STARLET.MLB;2                        10
TOTALS (all libraries)                                   11

99 GETS were required to define 11 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:HLDMAIN/OBJ=OBJ$:HLDMAIN MSRC$:HLDMAIN/UPDATE=(ENH$:HLDMAIN)+LIB$:HLD/LIB
```

0186 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



INIT

HLDPRINT
LIS

IMGDMPDEF
SDL

INIT
MAP

HLDDATA
LIS

INIBAD
LIS

INIDEF
B32

HLDMAIN
LIS

ANALIMDMP
LIS

IMGDMP

ANALIMDMP
MAP

IMGDMP
MAP

HLDIO
LIS

DISPIMDMP
LIS

HLDFILE
LIS

HLDTASK
LIS

IMGDMP
LIS

INIALL
LIS