```
   SSSSSSSSSSS  YYY         YYY    SSSSSSSSSSS  LLL              OOOOOOOOO       AAAAAAAAA
   SSSSSSSSSSS  YYY         YYY    SSSSSSSSSSS  LLL              OOOOOOOOO       AAAAAAAAA
   SSSSSSSSSSS  YYY         YYY    SSSSSSSSSSS  LLL              OOOOOOOOO       AAAAAAAAA
SSS             YYY         YYY SSS             LLL           OOO         OOO   AAA         AAA
SSS             YYY         YYY SSS             LLL           OOO         OOO   AAA         AAA
SSS             YYY         YYY SSS             LLL           OOO         OOO   AAA         AAA
SSS                YYY   YYY    SSS             LLL           OOO         OOO   AAA         AAA
SSS                YYY   YYY    SSS             LLL           OOO         OOO   AAA         AAA
SSS                YYY   YYY    SSS             LLL           OOO         OOO   AAA         AAA
   SSSSSSSSS          YYY          SSSSSSSSS     LLL           OOO         OOO   AAA         AAA
   SSSSSSSSS          YYY          SSSSSSSSS     LLL           OOO         OOO   AAA         AAA
   SSSSSSSSS          YYY          SSSSSSSSS     LLL           OOO         OOO   AAA         AAA
            SSS       YYY                   SSS  LLL           OOO         OOO   AAAAAAAAAAAAAAA
            SSS       YYY                   SSS  LLL           OOO         OOO   AAAAAAAAAAAAAAA
            SSS       YYY                   SSS  LLL           OOO         OOO   AAAAAAAAAAAAAAA
            SSS       YYY                   SSS  LLL           OOO         OOO   AAA         AAA
            SSS       YYY                   SSS  LLL           OOO         OOO   AAA         AAA
            SSS       YYY                   SSS  LLL           OOO         OOO   AAA         AAA
SSSSSSSSSSS           YYY SSSSSSSSSSS     LLLLLLLLLLLLLLL        OOOOOOOOO     AAA         AAA
SSSSSSSSSSS           YYY SSSSSSSSSSS     LLLLLLLLLLLLLLL        OOOOOOOOO     AAA         AAA
SSSSSSSSSSS           YYY SSSSSSSSSSS     LLLLLLLLLLLLLLL        OOOOOOOOO     AAA         AAA
```

FILEID**LIOSUBUV1

```
LL            IIIIII     000000     SSSSSSSS  UU      UU  BBBBBBBB    UU      UU  VV      VV     11
LL            IIIIII     000000     SSSSSSSS  UU      UU  BBBBBBBB    UU      UU  VV      VV     11
LL              II     OO      OO  SS        UU      UU  BB      BB  UU      UU  VV      VV   1111
LL              II     OO      OO  SS        UU      UU  BB      BB  UU      UU  VV      VV   1111
LL              II     OO      OO  SS        UU      UU  BB      BB  UU      UU  VV      VV     11
LL              II     OO      OO  SS        UU      UU  BB      BB  UU      UU  VV      VV     11
LL              II     OO      OO    SSSSSS  UU      UU  BBBBBBBB    UU      UU  VV      VV     11
LL              II     OO      OO    SSSSSS  UU      UU  BBBBBBBB    UU      UU  VV      VV     11
LL              II     OO      OO          SS  UU      UU  BB      BB  UU      UU  VV      VV     11
LL              II     OO      OO          SS  UU      UU  BB      BB  UU      UU  VV      VV     11
LL              II     OO      OO          SS  UU      UU  BB      BB  UU      UU    VV  VV       11
LL              II     OO      OO          SS  UU      UU  BB      BB  UU      UU    VV  VV       11
LLLLLLLLLL    IIIIII     000000     SSSSSSSS    UUUUUUUUUU  BBBBBBBB    UUUUUUUUUU      VV     111111
LLLLLLLLLL    IIIIII     000000     SSSSSSSS    UUUUUUUUUU  BBBBBBBB    UUUUUUUUUU      VV     111111
```

```
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     1          .NOSHOW CONDITIONALS
0000     5
0000     9
0000    13
0000    17
0000    19          .TITLE  LIOSUBUV1 - LOADABLE I/O SUBROUTINES
0000    21
0000    22          .IDENT  'V04-000'
0000    23
0000    24 ;
0000    25 ;*****************************************************************************
0000    26 ;*                                                                           *
0000    27 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    28 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000    29 ;*  ALL RIGHTS RESERVED.                                                     *
0000    30 ;*                                                                           *
0000    31 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    32 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    33 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    34 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    35 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    36 ;*  TRANSFERRED.                                                             *
0000    37 ;*                                                                           *
0000    38 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    39 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    40 ;*  CORPORATION.                                                             *
0000    41 ;*                                                                           *
0000    42 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    43 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    44 ;*                                                                           *
0000    45 ;*                                                                           *
0000    46 ;*****************************************************************************
0000    47 ;
0000    48 ;++
0000    49 ;
0000    50 ; FACILITY:
0000    51 ;
0000    52 ;       EXECUTIVE, I/O CONTROL ROUTINES
0000    53 ;
0000    54 ; ABSTRACT:
0000    55 ;
0000    56 ; I/O SUBROUTINES WHICH CONTAIN PROCESSOR DEPENDENCIES.
0000    57 ;
0000    58 ; AUTHOR:
0000    59 ;
0000    60 ;       N. KRONENBERG, JANUARY 12, 1979.
0000    61 ;
0000    62 ; MODIFIED BY:
0000    63 ;
0000    64 ;       V03-012 KDM0096         Kathleen D. Morse       27-Mar-1984
0000    65 ;               Add memory CSR scanning to IOC$PURGDATAP for MicroVAX I.
0000    66 ;               (All DMA MicroVAX I drivers should call this routine, just
0000    67 ;               before calling IOC$REQCOM.)
0000    68 ;
0000    69 ;       V03-011 KDM0081         Kathleen D. Morse       13-Sep-1983
0000    70 ;               Create a version for Micro-VAX I.
0000    71 ;
```

```
0000     72 ;       V03-010 TCM0004         Trudy C. Matthews       4-Jan-1982
0000     73 ;               Added 11/790-specific path to IOC$PURGDATAP.
0000     74 ;
0000     75 ;       V09     TCM0003         Trudy C. Matthews       9-Nov-1982
0000     76 ;               Added a .TITLE statement for LIOSUB790.
0000     77 ;
0000     78 ;       V08     TCM0002         Trudy C. Mathews        29-Jul-1981
0000     79 ;               Changed all "7ZZ"s to "730"s.
0000     80 ;
0000     81 ;       V07     TCM0001         Trudy C Matthews        28-Feb-1980
0000     82 ;               Changed IOC$PURGDATAP for NEBULA so that it logs
0000     83 ;               the Unibus Error Summary register itself when there
0000     84 ;               are Unibus errors reported.
0000     85 ;
0000     86 ;       V06     NPK0002         N. KRONENBERG           4-DEC-1979
0000     87 ;               REPLACED IOC$PURGDATAP FOR NEBULA
0000     88 ;
0000     89 ;       V05     NPK0001         N. KRONENBERG           23-AUG-1979
0000     90 ;               CORRECTED 11/750 CHECK FOR PURGE DONE.
0000     91 ;
0000     92 ;       V04     TCM0001         Trudy C. Matthews       3-Jul-1979
0000     93 ;               Modified IOC$PURGDATAP for NEBULA.
0000     94 ;
0000     95 ;--
```

```
                        0000     97 ;
                        0000     98 ; MACRO LIBRARY CALLS:
                        0000     99 ;
                        0000    100         $ADPDEF                         ; Define ADP offsets
                        0000    101         $CRBDEF                         ; Define CRB offsets
                        0000    102         $EMBETDEF                       ; Define error types.
                        0000    103         $EMBUEDEF                       ; Define Unibus Error buffer.
                        0000    104         $IDBDEF                         ; Define IDB offsets
                        0000    105         $PRDEF                          ; Define IPR'S
                        0000    106         $UBADEF                         ; Define UBA offsets
                        0000    107         $UBIDEF                         ; Define UBI offsets
                        0000    108         $UCBDEF                         ; Define UCB offsets
                        0000    109         $VECDEF                         ; Define CRB/VEC offsets
                        0000    110
                        0000    115
                        0000    120
                        0000    125
                        0000    130
00000000                0000    132         C780_LIKE = 0
00000001                0000    133         C750_LIKE = 1
                        0000    135
```

```
                                0000   137          .SBTTL  PURGE DATAPATH
                                0000   138 ;+
                                0000   139 ; IOC$PURGDATAP - PURGE DATAPATH
                                0000   140 ;
                                0000   141 ; This routine purges the caller's buffered datapath, and clears any
                                0000   142 ; datapath errors.  if there was a datapath error, this fact is
                                0000   143 ; returned to the caller.
                                0000   144 ;
                                0000   145 ; INPUTS:
                                0000   146 ;
                                0000   147 ;       R5 = UCB address
                                0000   148 ;
                                0000   149 ; OUTPUTS:
                                0000   150 ;
                                0000   151 ;       R0-R3 altered
                                0000   152 ;       Other registers preserved
                                0000   153 ;       R0 = low bit clear/set if transmission error/success
                                0000   154 ;       R1 = DPR contents after purge (for register dump by caller)
                                0000   155 ;       R2 = address of start of adapter map registers (for reg dump by caller)
                                0000   156 ;       R3 = CRB address
                                0000   157 ;-
                                0000   158
                           00000000   159          .PSECT  WIONONPAGED
                                0000   160
                                0000   161          .ENABL  LSB
                                0000   162
                                0000   163 IOC$PURGDATAP::                       ;
                                0000   185
                                0000   186
                                0000   214
                                0000   263
                                0000   265 ;
                                0000   266 ; The Micro-VAX I has no real datapath registers, so no actual
                                0000   267 ; purge is done.
                                0000   268 ;
                                0000   269 ; However, every DMA driver for MicroVAX I must call this routine
                                0000   270 ; just before calling IOC$REQCOM.  This is to cause the memory CSRs
                                0000   271 ; to be logged if a parity error occurred.  If this routine is not
                                0000   272 ; called by the driver, then the error log will probably not contain
                                0000   273 ; enough information to identify which memory controller took the error.
                                0000   274 ;
51   00000000'GF    DE          0000   275          MOVAL   G^EXE$AL_MEMCSRS,R1     ; Get address of memory CSR array.
          50    81  D0          0007   276          MOVL    (R1)+,R0                ; Get count of memory CSRs.
 00 B1    1F40  8F  B3          000A   277 10$:     BITW    #8000,@(R1)             ; Is parity error bit set in CSR?
                19  12          0010   278          BNEQ    30$                     ; Br if parity error occurred, go log it
          51    04  C0          0012   279          ADDL    #4,R1                   ; Get VA of next memory controller CSR.
            F2  50  F5          0015   280          SOBGTR  R0,10$                  ; Loop through all CSRs on system.
                                0018   281
       53   24  A5  D0          0018   282 20$:     MOVL    UCB$L_CRB(R5),R3        ; Get CRB address
       52   38  B3  D0          001C   283          MOVL    @CRB$L_INTD+VEC$L_ADP(R3),R2 ; Get start of adapter register space
    52    0800  C2  DE          0020   284          MOVAL   UBI$L_MAP(R2),R2        ; Return addr of 1st map reg.
                51  D4          0025   285          CLRL    R1                      ; No DPR contents.
          50    01  D0          0027   286          MOVL    #1,R0                   ; Set success always.
                05  05          002A   287          RSB                             ; Return
                                002B   288
                                002B   289 30$:                                     ; Memory parity error occurred.
          7E    54  7D          002B   290          MOVQ    R4,-(SP)                ; Save R4/R5.  R0-R3 wiped out.
```

```
                53    06    3C  002E    291          MOVZWL  #EMB$K_SE,R3            ; Set indicator to scan for errors.
00000000'GF     54    16    31  0031    292          JSB     G^EXE$LOGMEM            ; Log the memory error and bugcheck
                      8E    7D  0037    293          MOVQ    (SP)+,R4                ;  if necessary to maintain system
                                0039    294                                          ;  or data integrity.
                      DC    11  003A    295          BRB     20$                     ; Join common code.
                                003C    297
                                003C    298          .DSABL  LSB
                                003C    299
                                003C    300          .END
```

```
C750_LIKE                       = 00000001
C780_LIKE                       = 00000000
CPU_TYPE                        = 00000007
CRB$L_INTD                      = 00000024
EMB$K_SE                        = 00000006
EXE$AL_MEMCSRS                    ********    X   02
EXE$LOGMEM                        ********    X   02
IOC$PURGDATAP                     00000000 RG     02
PR$_SID_TYP730                  = 00000003
PR$_SID_TYP750                  = 00000002
PR$_SID_TYP780                  = 00000001
PR$_SID_TYP790                  = 00000004
PR$_SID_TYPUV1                  = 00000007
UBI$L_MAP                       = 00000800
UCB$L_CRB                       = 00000024
VEC$L_ADP                       = 00000014
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
WIONONPAGED	0000003C (60.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.05	00:00:00.88
Command processing	111	00:00:00.46	00:00:02.58
Pass 1	269	00:00:04.99	00:00:21.02
Symbol table sort	0	00:00:00.79	00:00:04.21
Pass 2	46	00:00:00.99	00:00:04.47
Symbol table output	4	00:00:00.03	00:00:00.57
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	462	00:00:07.32	00:00:33.74

```
The working set limit was 1350 pages.
44212 bytes (87 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 775 non-local and 3 local symbols.
304 source lines were read in Pass 1, producing 13 object records in Pass 2.
17 pages of virtual memory were used to define 16 macros.
```

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

Macro library name                    Macros defined
------------------                    --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                     9
_$255$DUA28:[SYSLIB]STARLET.MLB;2                  4
TOTALS (all libraries)                            13

864 GETS were required to define 13 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:LIOSUBUV1/OBJ=OBJ$:LIOSUBUV1 MSRC$:CPUSWUV1/UPDATE=(ENH$:CPUSWUV1)+MSRC$:LIOSUB/UPDATE=(ENH$:LIOSUB)+EXECML$/LIB
```

LIOSUB750
LIS

MCHECKUV1
LIS

MCHECK730
LIS

MCHECK750
LIS

LIOSUB730
LIS

LIOSUB790
LIS

MCHECK780
LIS

MCF790
LIS

MCHECK790
LIS

LIOSUB780
LIS

LIOSUBUV1
LIS