```
   SSSSSSSSSSS  YYY         YYY     SSSSSSSSSSS  LLL                000000000        AAAAAAAAA
   SSSSSSSSSSS  YYY         YYY     SSSSSSSSSSS  LLL                000000000        AAAAAAAAA
   SSSSSSSSSSS  YYY         YYY     SSSSSSSSSSS  LLL                000000000        AAAAAAAAA
SSS             YYY         YYY  SSS             LLL             000         000  AAA         AAA
SSS             YYY         YYY  SSS             LLL             000         000  AAA         AAA
SSS             YYY         YYY  SSS             LLL             000         000  AAA         AAA
SSS                YYY   YYY     SSS             LLL             000         000  AAA         AAA
SSS                YYY   YYY     SSS             LLL             000         000  AAA         AAA
SSS                YYY   YYY     SSS             LLL             000         000  AAA         AAA
   SSSSSSSSS          YYY          SSSSSSSSS     LLL             000         000  AAA         AAA
   SSSSSSSSS          YYY          SSSSSSSSS     LLL             000         000  AAA         AAA
   SSSSSSSSS          YYY          SSSSSSSSS     LLL             000         000  AAA         AAA
            SSS       YYY                   SSS  LLL             000         000  AAAAAAAAAAAAAAA
            SSS       YYY                   SSS  LLL             000         000  AAAAAAAAAAAAAAA
            SSS       YYY                   SSS  LLL             000         000  AAAAAAAAAAAAAAA
            SSS       YYY                   SSS  LLL             000         000  AAA         AAA
            SSS       YYY                   SSS  LLL             000         000  AAA         AAA
            SSS       YYY                   SSS  LLL             000         000  AAA         AAA
SSSSSSSSSSS           YYY     SSSSSSSSSSS     LLLLLLLLLLLLLLL     000000000     AAA         AAA
SSSSSSSSSSS           YYY     SSSSSSSSSSS     LLLLLLLLLLLLLLL     000000000     AAA         AAA
SSSSSSSSSSS           YYY     SSSSSSSSSSS     LLLLLLLLLLLLLLL     000000000     AAA         AAA
```

```
LL          IIIIII    000000    SSSSSSSS  UU      UU  BBBBBBBB  77777777    333333    000000
LL          IIIIII    000000    SSSSSSSS  UU      UU  BBBBBBBB  77777777    333333    000000
LL            II    00      00  SS        UU      UU  BB      BB        77  33      33  00      00
LL            II    00      00  SS        UU      UU  BB      BB        77  33      33  00      00
LL            II    00      00  SS        UU      UU  BB      BB        77          33  00    0000
LL            II    00      00  SS        UU      UU  BB      BB        77          33  00    0000
LL            II    00      00    SSSSSS  UU      UU  BBBBBBBB        77          33  00  00  00
LL            II    00      00    SSSSSS  UU      UU  BBBBBBBB        77          33  00  00  00
LL            II    00      00        SS  UU      UU  BB      BB      77          33  0000    00
LL            II    00      00        SS  UU      UU  BB      BB      77          33  0000    00
LL            II    00      00        SS  UU      UU  BB      BB    77        33    33  00      00
LL            II    00      00        SS  UU      UU  BB      BB    77        33    33  00      00
LLLLLLLLLL  IIIIII    000000  SSSSSSSS    UUUUUUUUUU  BBBBBBBB  77          333333    000000
LLLLLLLLLL  IIIIII    000000  SSSSSSSS    UUUUUUUUUU  BBBBBBBB  77          333333    000000

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
```

```
0000          .NOSHOW CONDITIONALS

0000          .TITLE  LIOSUB730 - LOADABLE I/O SUBROUTINES


0000          .IDENT  'V04-000'

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:
0000  ;
0000  ;       EXECUTIVE, I/O CONTROL ROUTINES
0000  ;
0000  ; ABSTRACT:
0000  ;
0000  ; I/O SUBROUTINES WHICH CONTAIN PROCESSOR DEPENDENCIES.
0000  ;
0000  ; AUTHOR:
0000  ;
0000  ;       N. KRONENBERG, JANUARY 12, 1979.
0000  ;
0000  ; MODIFIED BY:
0000  ;
0000  ;       V03-012 KDM0096         Kathleen D. Morse       27-Mar-1984
0000  ;               Add memory CSR scanning to IOC$PURGDATAP for MicroVAX I.
0000  ;               (All DMA MicroVAX I drivers should call this routine, just
0000  ;               before calling IOC$REQCOM.)
0000  ;
0000  ;       V03-011 KDM0081         Kathleen D. Morse       13-Sep-1983
0000  ;               Create a version for Micro-VAX I.
0000  ;
```

```
0000    ;       V03-010 TCM0004         Trudy C. Matthews       4-Jan-1982
0000    ;               Added 11/790-specific path to IOC$PURGDATAP.
0000    ;
0000    ;       V09     TCM0003         Trudy C. Matthews       9-Nov-1982
0000    ;               Added a .TITLE statement for LIOSUB790.
0000    ;
0000    ;       V08     TCM0002         Trudy C. Mathews        29-Jul-1981
0000    ;               Changed all "7ZZ"s to "730"s.
0000    ;
0000    ;       V07     TCM0001         Trudy C Matthews        28-Feb-1980
0000    ;               Changed IOC$PURGDATAP for NEBULA so that it logs
0000    ;               the Unibus Error Summary register itself when there
0000    ;               are Unibus errors reported.
0000    ;
0000    ;       V06     NPK0002         N. KRONENBERG           4-DEC-1979
0000    ;               REPLACED IOC$PURGDATAP FOR NEBULA
0000    ;
0000    ;       V05     NPK0001         N. KRONENBERG           23-AUG-1979
0000    ;               CORRECTED 11/750 CHECK FOR PURGE DONE.
0000    ;
0000    ;       V04     TCM0001         Trudy C. Matthews       3-Jul-1979
0000    ;       Modified IOC$PURGDATAP for NEBULA.
0000    ;
0000    ;--
```

```
                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
00000000        0000   122          C780_LIKE = 0
00000001        0000   123          C750_LIKE = 1
                0000   125
                0000   130
                0000   135
```

```
                 0000          .SBTTL  PURGE DATAPATH
                 0000   ;+
                 0000   ; IOC$PURGDATAP - PURGE DATAPATH
                 0000   ;
                 0000   ; This routine purges the caller's buffered datapat   and clears any
                 0000   ; datapath errors.  if there was a datapath error, tn s fact is
                 0000   ; returned to the caller.
                 0000   ;
                 0000   ; INPUTS:
                 0000   ;
                 0000   ;       R5 = UCB address
                 0000   ;
                 0000   ; OUTPUTS:
                 0000   ;
                 0000   ;       R0-R3 altered
                 0000   ;       Other registers preserved
                 0000   ;       R0 = low bit clear/set if transmission error/success
                 0000   ;       R1 = DPR contents after purge (for register dump by caller)
                 0000   ;       R2 = address of start of adapter map registers (for reg dump by caller)
                 0000   ;       R3 = CRB address
                 0000   ;-
                 0000
         00000000        .PSECT  WIONONPAGED
                 0000
                 0000          .ENABL  LSB
                 0000
                 0000   IOC$PURGDATAP::                         ;
                 0000
                 0000
                 0000
                 0000   ;
                 0000   ; The VAX 11/730 has no real datapath registers, so no actual
                 0000   ; purge is done.  The 11/730 maintains a shared UNIBUS error
                 0000   ; register, which is checked here.  if an error is shown
                 0000   ; in the shared error register, it is logged as a UNIBUS error.
                 0000   ; note, however, that since the error register is shared, the error
                 0000   ; could possibly belong to another device doing a transfer
                 0000   ; concurrently.  the error will show up twice in the errorlog; once
                 0000   ; as a UNIBUS error and once as an NXM in a device register for the
                 0000   ; device that really had the error.
                 0000   ;
                 0000
          10 BB  0000          PUSHR   #^M<R4>                 ; Save register
     53   24 A5 D0  0002       MOVL    UCB$L_CRB(R5),R3        ; Get CRB address
     52   38 B3 D0  0006       MOVL    @CRB$L_INTD+VEC$L_ADP(R3),R2 ; Get start of adapter register space
                 000A
          00 EF  000A          EXTZV   #VEC$V_DATAPATH,-       ; Extract datapath #
          05     000C                  #VEC$S_DATAPATH,-       ;  from CRB
     51   37 A3  000D                  CRB$L_INTD+VEC$B_DATAPATH(R3),R1
     54   10 A2 D0  0010       MOVL    UBI$L_SR(R2),R4         ; Get Unibus Error Summary reg.
54   8001C000 8F D3  0014      BITL    #<UBI$M_SR_UWE!-        ; Any UB errors? (write error,
                 001B                  UBI$M_SR_MRPE!-         ;       map parity error,
                 001B                  UBI$M_SR_NXM!-          ;       non-existent memory,
                 001B                  UBI$M_SR_UCE>,R4        ; or uncorrected read error.)
          1E 13  001B          BEQL    30$                     ; Branch if no errors.
                 001D   ;
                 001D   ; The shared Unibus error register indicates an error.  Log it.
```

```
                    001D   ;
          52   DD   001D          PUSHL   R2                          ; Save addr of adaptor reg space
     51   14   9A   001F          MOVZBL  #EMB$C_UE_LENGTH,R1         ; Set length of error buffer.
00000000'GF   16   0022          JSB     G^ERL$ALLOCEMB              ; Get a message buffer.
        0E 50   E9   0028          BLBC    R0,20$                      ; Branch if unsuccessful.
  10 A2   54   D0   002B          MOVL    R4,EMB$L_UE_UBERR(R2)       ; Put error register in buffer.
  04 A2   0B   9A   002F          MOVZBL  #EMB$C_UE, -                ; Set entry type in buffer.
                    0033                    EMB$W_UE_ENTRY(R2)          ;
00000000'GF   16   0033          JSB     G^ERL$RELEASEMB             ; Release buffer to errorlogger.
          04   BA   0039   20$:   POPR    #^M<R2>                     ; Restore addr of adaptor regs.
                    003B   ;
                    003B   ; Set up outputs from PURGDATAP.
                    003B   ;
     50   01   D0   003B   30$:   MOVL    #1,R0                       ; Set success always.
          51   D4   003E          CLRL    R1                          ; No DPR contents.
52   0800 C2   DE   0040          MOVAL   UBI$L_MAP(R2),R2            ; Return addr of 1st map reg.
                    0045
          10   BA   0045          POPR    #^M<R4>                     ; Restore register
               05   0047          RSB                                 ; Return
                    0048
                    0048
                    0048          .DSABL  LSB
                    0048
                    0048          .END
```

```
C750_LIKE                       = 00000001
C780_LIKE                       = 00000000
CPU_TYPE                        = 00000003
CRB$L_INTD                      = 00000024
EMB$C_UE                        = 0000000B
EMB$C_UE_LENGTH                 = 00000014
EMB$L_UE_UBERR                  = 00000010
EMB$W_UE_ENTRY                  = 00000004
ERL$ALLOCEMB                      ********   X   02
ERL$RELEASEMB                     ********   X   02
IOC$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
UBI$L_SR                        = 00000010
UBI$M_SR_MRPE                   = 00008000
UBI$M_SR_NXM                    = 00010000
UBI$M_SR_UCE                    = 80000000
UBI$M_SR_UWE                    = 00004000
UCB$L_CRB                       = 00000024
VEC$B_DATAPATH                  = 00000013
VEC$L_ADP                       = 00000014
VEC$S_DATAPATH                  = 00000005
VEC$V_DATAPATH                  = 00000000
```

+----------+
! 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	00000048	(72.)	02	(2.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.05	00:00:01.87
Command processing	137	00:00:00.48	00:00:02.40
Pass 1	279	00:00:04.97	00:00:17.76
Symbol table sort	0	00:00:00.79	00:00:02.00
Pass 2	49	00:00:00.99	00:00:04.76
Symbol table output	5	00:00:00.03	00:00:00.03
Psect synopsis output	2	00:00:00.01	00:00:00.30
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	509	00:00:07.33	00:00:29.12

The working set limit was 1350 pages.
44149 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 2 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$:LIOSUB730/OBJ=OBJ$:LIOSUB730 MSRC$:CPUSW730/UPDATE=(ENH$:CPUSW730)+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