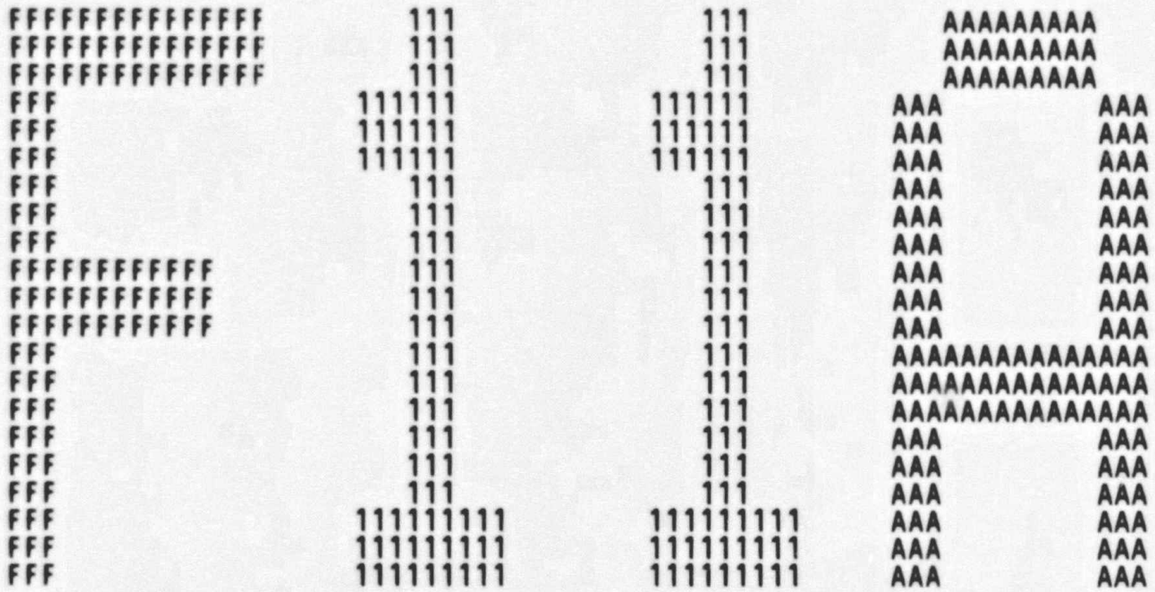
FFFFFFFFFFFFFFF
FFFFFFFFFFFFFFF
FFFFFFFFFFFFFFF
FFF
FFF
FFF
FFF
FFF
FFF
FFFFFFFFFFFF
FFFFFFFFFFFF
FFFFFFFFFFFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
111
111
111
111111
111111
111111
111
111
111
111
111
111
111
111
111
111
111
111111111
111111111
111111111
111
111
111
111111
111111
111111
111
111
111
111
111
111
111
111
111
111
111
111111111
111111111
111111111
AAAAAAAAA
AAAAAAAAA
AAAAAAAAA
AAA AAA
AAA AAA
AAA AAA
AAA AAA
AAA AAA
AAA AAA
AAA AAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAA AAA
AAA AAA
AAA AAA
AAA AAA
AAA AAA
AAA AAA

```
  GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT  RRRRRRRR    EEEEEEEEEE    QQQQQQ
  GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT  RRRRRRRR    EEEEEEEEEE    QQQQQQ
GG          EE              TT      RR      RR  EE          QQ      QQ
GG          EE              TT      RR      RR  EE          QQ      QQ
GG          EE              TT      RR      RR  EE          QQ      QQ
GG          EE              TT      RR      RR  EE          QQ      QQ
GG          EEEEEEEE        TT      RRRRRRRR    EEEEEEEE    QQ      QQ
GG          EEEEEEEE        TT      RRRRRRRR    EEEEEEEE    QQ      QQ
GG  GGGGGG  EE              TT      RR  RR      EE          QQ  QQ  QQ
GG  GGGGGG  EE              TT      RR  RR      EE          QQ  QQ  QQ
GG      GG  EE              TT      RR    RR    EE          QQ    QQ        ....
GG      GG  EE              TT      RR    RR    EE          QQ    QQ        ....
  GGGGGG    EEEEEEEEEE      TT      RR      RR  EEEEEEEEEE    QQQQ  QQ      ....
  GGGGGG    EEEEEEEEEE      TT      RR      RR  EEEEEEEEEE    QQQQ  QQ      ....

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
```

```
;     1          0001  0 MODULE GETREQ (
;     2          0002  0                 LANGUAGE (BLISS32),
;     3          0003  0                 IDENT = 'V04-000'
;     4          0004  0                 ) =
;     5          0005  1 BEGIN
;     6          0006  1
;     7          0007  1 !
;     8          0008  1 !**********************************************************************************
;     9          0009  1 !*                                                                                *
;    10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                       *
;    11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                         *
;    12          0012  1 !*  ALL RIGHTS RESERVED.                                                           *
;    13          0013  1 !*                                                                                *
;    14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED          *
;    15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE         *
;    16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER         *
;    17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY         *
;    18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY         *
;    19          0019  1 !*  TRANSFERRED.                                                                   *
;    20          0020  1 !*                                                                                *
;    21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE         *
;    22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT         *
;    23          0023  1 !*  CORPORATION.                                                                   *
;    24          0024  1 !*                                                                                *
;    25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS         *
;    26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                        *
;    27          0027  1 !*                                                                                *
;    28          0028  1 !*                                                                                *
;    29          0029  1 !**********************************************************************************
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 !
;    33          0033  1 ! FACILITY:  F11ACP Structure Level 1
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !
;    37          0037  1 !       This routine gets the next I/O request from the ACP queue.
;    38          0038  1 !       If no requests are queued, it hibernates.
;    39          0039  1 !
;    40          0040  1 ! ENVIRONMENT:
;    41          0041  1 !
;    42          0042  1 !       STARLET operating system, including privileged system services
;    43          0043  1 !       and internal exec routines. This routine must be called
;    44          0044  1 !       in kernel mode.
;    45          0045  1 !
;    46          0046  1 !--
;    47          0047  1 !
;    48          0048  1 !
;    49          0049  1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  19-Dec-1976  17:26
;    50          0050  1 !
;    51          0051  1 ! MODIFIED BY:
;    52          0052  1 !
;    53          0053  1 !       V03-002 LMP0221         L. Mark Pilant,         27-Mar-1984  13:13
;    54          0054  1 !               Change UCB$L_OWNUIC to ORB$L_OWNER and UCB$W_VPROT to
;    55          0055  1 !               ORB$W_PROT.
;    56          0056  1 !
;    57          0057  1 !       V03-001 ACG0408         Andrew C. Goldstein,    20-Mar-1984  16:37
```

```
;    58          0058  1 !                     Reduce size of LOCAL_ARB
;    59          0059  1 !
;    60          0060  1 !       V02-004 ACG38100        Andrew C. Goldstein,    3-Jun-1981  11:45
;    61          0061  1 !               Fix granting of SYSPRV to volume owner
;    62          0062  1 !
;    63          0063  1 !       V02-003 ACG0167         Andrew C. Goldstein,    7-May-1980  18:50
;    64          0064  1 !               Previous revision history moved to F11A.REV
;    65          0065  1 !**
;    66          0066  1
;    67          0067  1
;    68          0068  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    69          0069  1 REQUIRE 'SRC$:FCPDEF.B32';
;    70          0384  1
;    71          0385  1
;    72          0386  1 FORWARD ROUTINE
;    73          0387  1         GET_REQUEST,                    ! get next FCP request
;    74          0388  1         GET_CCB;                        ! get address of CCB of channel
```

```
;    76          0389  1 GLOBAL ROUTINE GET_REQUEST =
;    77          0390  1
;    78          0391  1 !++
;    79          0392  1 !
;    80          0393  1 ! FUNCTIONAL DESCRIPTION:
;    81          0394  1 !
;    82          0395  1 !       This routine gets the next I/O request from the ACP queue.
;    83          0396  1 !       If no requests are queued, it hibernates.
;    84          0397  1 !
;    85          0398  1 ! CALLING SEQUENCE:
;    86          0399  1 !       GET_REQUEST ()
;    87          0400  1 !
;    88          0401  1 ! INPUT PARAMETERS:
;    89          0402  1 !       NONE
;    90          0403  1 !
;    91          0404  1 ! IMPLICIT INPUTS:
;    92          0405  1 !       QUEUE_HEAD: address of ACP queue block
;    93          0406  1 !       IO_CHANNEL: I/O channel number
;    94          0407  1 !
;    95          0408  1 ! OUTPUT PARAMETERS:
;    96          0409  1 !       NONE
;    97          0410  1 !
;    98          0411  1 ! IMPLICIT OUTPUTS:
;    99          0412  1 !       CURRENT_UCB: address of UCB of request
;   100          0413  1 !       CURRENT_VCB: address of VCB of request
;   101          0414  1 !       CURRENT_WINDOW: window of file if accessed
;   102          0415  1 !       PRIMARY_FCB: FCB of file if accessed
;   103          0416  1 !
;   104          0417  1 ! ROUTINE VALUE:
;   105          0418  1 !       address of request I/O packet
;   106          0419  1 !
;   107          0420  1 ! SIDE EFFECTS:
;   108          0421  1 !       I/O channel assigned to device of request
;   109          0422  1 !
;   110          0423  1 !--
;   111          0424  1
;   112          0425  2 BEGIN
;   113          0426  2
;   114          0427  2 ! Note that the ACP queue header must be referenced through an explicit
;   115          0428  2 ! register. This is to ensure that the REMQUE expression in fact produces
;   116          0429  2 ! an single REMQUE instruction, and is not broken into an instruction
;   117          0430  2 ! sequence that is not interlocked.
;   118          0431  2 !
;   119          0432  2 REGISTER
;   120          0433  2         QUEUE_POINTER   : REF BBLOCK;
;   121          0434  2
;   122          0435  2 LOCAL
;   123          0436  2         P               : REF BBLOCK,   ! pointer to chase AQB list
;   124          0437  2         CCB             : REF BBLOCK,   ! pointer to CCB of i/o channel
;   125          0438  2         ORB             : REF BBLOCK,   ! Pointer to ORB from UCB
;   126          0439  2         ABD             : REF BBLOCKVECTOR [,ABD$C_LENGTH],
;   127          0440  2                                         ! pointer to buffer descriptor packet
;   128          0441  2         PACKET          : REF BBLOCK;   ! address of new I/O packet
;   129          0442  2
;   130          0443  2 EXTERNAL
;   131          0444  2         CLEANUP_FLAGS   : BITVECTOR,    ! cleanup action flags
;   132          0445  2         QUEUE_HEAD      : REF BBLOCK,   ! ACP queue list head
```

```
;  133     0446  2          DISK_UCB        : REF BBLOCK,    ! UCB of device 'SYS$DISK'
;  134     0447  2          CURRENT_UCB     : REF BBLOCK,    ! address of current UCB
;  135     0448  2          CURRENT_VCB     : REF BBLOCK,    ! address of current VCB
;  136     0449  2          CURRENT_WINDOW  : REF BBLOCK,    ! address of file window
;  137     0450  2          PRIMARY_FCB     : REF BBLOCK,    ! address of file FCB
;  138     0451  2          LOCAL_ARB       : BBLOCK,        ! local copy of caller's ARB
;  139     0452  2          IO_CHANNEL,                      ! channel for all I/O
;  140     0453  2          EXE$GL_SYSUIC   : ADDRESSING_MODE (ABSOLUTE),
;  141     0454  2                                           ! highest SYSTEM UIC
;  142     0455  2          IOC$GL_AQBLIST  : REF BBLOCK ADDRESSING_MODE (ABSOLUTE);
;  143     0456  2                                           ! system AQB listhead
;  144     0457  2
;  145     0458  2      EXTERNAL ROUTINE
;  146     0459  2              LOCK_IODB,                   ! interlock system I/O database
;  147     0460  2              UNLOCK_IODB,                 ! unlock system I/O database
;  148     0461  2              DEALLOCATE;                  ! deallocate system dynamic memory
;  149     0462  2
;  150     0463  2      ! Attempt to dequeue a packet. If unsuccessful, hibernate and try again.
;  151     0464  2      !
;  152     0465  2
;  153     0466  2      WHILE 1 DO
;  154     0467  2          BEGIN
;  155     0468  2          QUEUE_POINTER = .QUEUE_HEAD;
;  156     0469  2          IF NOT REMQUE (.QUEUE_POINTER[AQB$L_ACPQFL], PACKET)
;  157     0470  2          THEN EXITLOOP;
;  158     0471  2
;  159     0472  2          IF .QUEUE_POINTER[AQB$B_MNTCNT] EQL 0
;  160     0473  2          THEN
;  161     0474  3
;  162     0475  3      ! If the REMQUE failed and the mount count in the AQB is zero, this ACP is
;  163     0476  3      ! potentially idle. Interlock the I/O database and check the queue and the
;  164     0477  3      ! count again. If the ACP is no longer idle, proceed as if nothing had happened.
;  165     0478  3      ! If it still is, unhook the AQB from the system AQB list. Once unhooked, the
;  166     0479  3      ! ACP can no longer be found by anyone. Then restore the original UCB of our
;  167     0480  3      ! I/O channel, deassign the channel, and go away.
;  168     0481  3      !
;  169     0482  3
;  170     0483  4              BEGIN
;  171     0484  4              LOCK_IODB ();
;  172     0485  4              IF .QUEUE_POINTER[AQB$B_MNTCNT] EQL 0
;  173     0486  4              AND .QUEUE_POINTER[AQB$[_ACPQFL] EQL QUEUE_POINTER[AQB$L_ACPQFL]
;  174     0487  4              THEN
;  175     0488  5                  BEGIN
;  176     0489  5                  P = .IOC$GL_AQBLIST;
;  177     0490  5                  IF .P EQL .QUEUE_POINTER
;  178     0491  5                  THEN IOC$GL_AQBLIST = .QUEUE_POINTER[AQB$L_LINK]
;  179     0492  5                  ELSE
;  180     0493  6                      BEGIN
;  181     0494  6                      UNTIL .P[AQB$L_LINK] EQL .QUEUE_POINTER
;  182     0495  6                      DO P = .P[AQB$[_LINK];
;  183     0496  6                      P[AQB$L_LINK] = .QUEUE_POINTER[AQB$L_LINK];
;  184     0497  5                      END;
;  185     0498  5                  UNLOCK_IODB ();
;  186     0499  5                  DEALLOCATE (.QUEUE_POINTER);
;  187     0500  5                  CCB = GET_CCB (.IO_CHANNEL);
;  188     0501  5                  CCB[CCB$L_UCB] = .DISK_UCB;
;  189     0502  5                  $DASSGN (CHAN = .IO_CHANNEL);
```

```
;  190     0503  5              $DELPRC ();
;  191     0504  5              END
;  192     0505  4          ELSE
;  193     0506  4              UNLOCK_IODB ();
;  194     0507  3          END;
;  195     0508  3
;  196     0509  3      $HIBER;
;  197     0510  2      END;                                    ! end of ACP wait loop
;  198     0511  2
;  199     0512  2  ! First check the type code in the packet.
;  200     0513  2  !
;  201     0514  2
;  202     0515  2  IF .PACKET[IRP$B_TYPE] NEQ DYN$C_IRP
;  203     0516  2  THEN BUG_CHECK (NOTIRPAQB, FATAL, 'Not IRP pointer in AQB');
;  204     0517  2
;  205     0518  2  ! Set up the UCB and VCB pointers and assign the I/O channel to the UCB.
;  206     0519  2  ! Check the type codes on all packets and control blocks.
;  207     0520  2  !
;  208     0521  2
;  209     0522  2  CURRENT_UCB = .PACKET[IRP$L_UCB];
;  210     0523  2  IF .CURRENT_UCB[UCB$B_TYPE] NEQ DYN$C_UCB
;  211     0524  2  THEN BUG_CHECK (NOTUCBIRP, FATAL, 'Not UCB pointer in IRP');
;  212     0525  2
;  213     0526  2  CURRENT_VCB = .CURRENT_UCB[UCB$L_VCB];
;  214     0527  2  IF .CURRENT_VCB[VCB$B_TYPE] NEQ DYN$C_VCB
;  215     0528  2  THEN BUG_CHECK (NOTVCBUCB, FATAL, 'Not VCB pointer in UCB');
;  216     0529  2
;  217     0530  2  CCB = GET_CCB (.IO_CHANNEL);
;  218     0531  2  CCB[CCB$L_UCB] = .CURRENT_UCB;              ! and assign it by stuffing UCB
;  219     0532  2  ORB = .CURRENT_UCB[UCB$L_ORB];
;  220     0533  2
;  221     0534  2  ! Get the window and FCB addresses if there is a file open on the channel.
;  222     0535  2  ! If the low bit of the window pointer is on, ignore the pointer (deaccess pending).
;  223     0536  2  !
;  224     0537  2
;  225     0538  2  CURRENT_WINDOW = .PACKET[IRP$L_WIND];
;  226     0539  2  IF .(PACKET[IRP$L_WIND])<0,1>
;  227     0540  2  THEN CURRENT_WINDOW = 0;
;  228     0541  2  IF .(PACKET[IRP$L_WIND])<1,2> NEQ 0
;  229     0542  2  THEN BUG_CHECK (BADWCBPT, FATAL, 'Bad WCB pointer in IRP');
;  230     0543  2
;  231     0544  2  IF .CURRENT_WINDOW NEQ 0
;  232     0545  2  THEN
;  233     0546  3      BEGIN
;  234     0547  3      IF .CURRENT_WINDOW[WCB$B_TYPE] NEQ DYN$C_WCB
;  235     0548  3      THEN BUG_CHECK (NOTWCBIRP, FATAL, 'Not WCB Pointer in IRP');
;  236     0549  3
;  237     0550  3      IF .CURRENT_WINDOW[WCB$V_NOTFCP]
;  238     0551  3      THEN BUG_CHECK (NOTFCPWCB, FATAL, 'Not FCP window in IRP');
;  239     0552  3
;  240     0553  3      PRIMARY_FCB = .CURRENT_WINDOW[WCB$L_FCB];
;  241     0554  3      IF .PRIMARY_FCB[FCB$B_TYPE] NEQ DYN$C_FCB
;  242     0555  3      THEN BUG_CHECK (NOTFCBWCB, FATAL, 'Bad FCB pointer in window');
;  243     0556  2      END;
;  244     0557  2
;  245     0558  2  ! If this is a normal file processor request (as opposed to a window turn),
;  246     0559  2  ! clear the byte count in the descriptor for the channel window
```

```
;  247          0560  2 ! pointer to inhibit write-back. Also zero out the result string and
;  248          0561  2 ! length buffers. Set the spool file bit is this is I/O to a spool file.
;  249          0562  2 ! This is denoted for ACP functions by noting that IRP$L_UCB is different
;  250          0563  2 ! from IRP$L_MEDIA (the latter containing the spooled device UCB address.
;  251          0564  2 !
;  252          0565  2
;  253          0566  2 IF .PACKET[IRP$V_COMPLX]
;  254          0567  2 THEN
;  255          0568  3     BEGIN
;  256          0569  3     ABD = .BBLOCK [.PACKET[IRP$L_SVAPTE], AIB$L_DESCRIPT];
;  257          0570  3     ABD[ABD$C_WINDOW, ABD$W_COUNT] = 0;
;  258          0571  3     IF .PACKET[IRP$L_UCB] NEQ .PACKET[IRP$L_MEDIA]
;  259          0572  3     THEN CLEANUP_FLAGS[CLF_SPOOLFILE] = 1;
;  260          0573  3     END
;  261          0574  3
;  262          0575  3 ! If there is no buffer packet, the function must be an ACP control function.
;  263          0576  3 !
;  264          0577  3
;  265          0578  2 ELSE
;  266          0579  3     BEGIN
;  267          0580  3     IF .PACKET[IRP$V_FCODE] GTRU IO$_LOGICAL
;  268          0581  3     AND .PACKET[IRP$V_FCODE] NEQ IO$_ACPCONTROL
;  269          0582  3     THEN BUG_CHECK (NOBUFPCKT, FATAL, 'Required buffer packet not present');
;  270          0583  2     END;
;  271          0584  2
;  272          0585  2 ! Set the system privilege flag bit, based on the caller's UIC and
;  273          0586  2 ! privileges.
;  274          0587  2 !
;  275          0588  2
;  276          0589  2 CH$MOVE (ARB$C_HEADER, .PACKET[IRP$L_ARB], LOCAL_ARB);
;  277          0590  2 IF .BBLOCK [LOCAL_ARB[ARB$Q_PRIV], PRV$V_SYSPRV]
;  278          0591  2 OR .BBLOCK [LOCAL_ARB[ARB$Q_PRIV], PRV$V_BYPASS]
;  279          0592  2 OR .(LOCAL_ARB[ARB$L_UIC])<16,16> LEQU .EXE$GL_SYSUIC
;  280          0593  2 OR .LOCAL_ARB[ARB$L_UIC] EQL .ORB[ORB$L_OWNER]
;  281          0594  2 THEN
;  282          0595  3     BEGIN
;  283          0596  3     CLEANUP_FLAGS[CLF_SYSPRV] = 1;
;  284          0597  3     BBLOCK [LOCAL_ARB[ARB$Q_PRIV], PRV$V_SYSPRV] = 1;
;  285          0598  2     END;
;  286          0599  2
;  287          0600  2 RETURN .PACKET;
;  288          0601  2
;  289          0602  1 END;                                    ! end of routine GETREQ


                                .TITLE  GETREQ
                                .IDENT  \V04-000\

                                .EXTRN  CLEANUP_FLAGS, QUEUE_HEAD
                                .EXTRN  DISK_UCB, CURRENT_UCB
                                .EXTRN  CURRENT_VCB, CURRENT_WINDOW
                                .EXTRN  PRIMARY_FCB, LOCAL_ARB
                                .EXTRN  IO_CHANNEL, EXE$GL_SYSUIC
                                .EXTRN  IOC$GL_AQBLIST, LOCK_IODB
                                .EXTRN  UNLOCK_IODB, DEALLOCATE
                                .EXTRN  SYS$DASSGN, SYS$DELPRC
                                .EXTRN  SYS$HIBER, BUG$_NOTIRPAQB
```

```
                                        .EXTRN  BUG$_NOTUCBIRP, BUG$_NOTVCBUCB
                                        .EXTRN  BUG$_BADWCBPT, BUG$_NOTWCBIRP
                                        .EXTRN  BUG$_NOTFCPWCB, BUG$_NOTFCBWCB
                                        .EXTRN  BUG$_NOBUFPCKT

                                        .PSECT  $CODE$,NOWRT,2

                        0FFC 00000      .ENTRY  GET_REQUEST, Save R2,R3,R4,R5,R6,R7,R8,R9,-      ; 0389
                                                R10,R11
                5B      0000G   CF  9E  00002   MOVAB   CURRENT_UCB, R11                         ;
                5A  00000000G   9F  9E  00007   MOVAB   @#IOC$GL_AQBLIST, R10                   ;
                59      0000G   CF  9E  0000E   MOVAB   CURRENT_WINDOW, R9                      ;
                58      0000G   CF  9E  00013   MOVAB   LOCAL_ARB, R8                           ;
                52      0000G   CF  D0  00018 1$:      MOVL    QUEUE_HEAD, QUEUE_POINTER       ; 0468
                56        00    B2  0F  0001D   REMQUE  @0(QUEUE_POINTER), PACKET               ; 0469
                                  77  1C  00021   BVC     7$                                      ;
                                  54  D4  00023   CLRL    R4                                      ; 0472
                            0B    A2  95  00025   TSTB    11(QUEUE_POINTER)                       ;
                                  66  12  00028   BNEQ    6$                                      ;
                                  54  D6  0002A   INCL    R4                                      ;
          0000G CF                00  FB  0002C   CALLS   #0, LOCK_IODB                         ; 0484
                57                54  E9  00031   BLBC    R4, 5$                                  ; 0485
                52                62  D1  00034   CMPL    (QUEUE_POINTER), QUEUE_POINTER          ; 0486
                                  52  12  00037   BNEQ    5$                                      ;
                53                6A  D0  00039   MOVL    IOC$GL_AQBLIST, P                       ; 0489
                52                53  D1  0003C   CMPL    P, QUEUE_POINTER                        ; 0490
                                  06  12  0003F   BNEQ    2$                                      ;
                6A          10    A2  D0  00041   MOVL    16(QUEUE_POINTER), IOC$GL_AQBLIST       ; 0491
                                  11  11  00045   BRB     4$                                      ;
                52          10    A3  D1  00047 2$:      CMPL    16(P), QUEUE_POINTER            ; 0494
                                  06  13  0004B   BEQL    3$                                      ;
                53          10    A3  D0  0004D   MOVL    16(P), P                                ; 0495
                                  F4  11  00051   BRB     2$                                      ;
              10    A3          10    A2  D0  00053 3$:      MOVL    16(QUEUE_POINTER), 16(P)        ; 0496
          0000G CF                00  FB  00058 4$:      CALLS   #0, UNLOCK_IODB                 ; 0498
                                  52  DD  0005D   PUSHL   QUEUE_POINTER                          ; 0499
          0000G CF                01  FB  0005F   CALLS   #1, DEALLOCATE
                        0000G     CF  DD  00064   PUSHL   IO_CHANNEL                            ; 0500
          0000V CF                01  FB  00068   CALLS   #1, GET_CCB
                55                50  D0  0006D   MOVL    R0, CCB
                65      0000G     CF  D0  00070   MOVL    DISK_UCB, (CCB)                         ; 0501
                        0000G     CF  DD  00075   PUSHL   IO_CHANNEL                            ; 0502
  00000000G     00                01  FB  00079   CALLS   #1, SYS$DASSGN
                                  7E  7C  00080   CLRQ    -(SP)                                   ; 0503
  00000000G     00                02  FB  00082   CALLS   #2, SYS$DELPRC
                                  05  11  00089   BRB     6$                                      ; 0485
          0000G CF                00  FB  0008B 5$:      CALLS   #0, UNLOCK_IODB                 ; 0506
  00000000G     00                00  FB  00090 6$:      CALLS   #0, SYS$HIBER                   ; 0507
                            FF7E  31  00097   BRW     1$                                      ; 0466
                0A          0A    A6  91  0009A 7$:      CMPB    10(PACKET), #10                 ; 0515
                                  04  13  0009E   BEQL    8$                                      ;
                                FEFF  000A0     BUGW                                            ; 0516
                              0000* 000A2       .WORD   <BUG$_NOTIRPAQB!4>
                6B          1C    A6  D0  000A4 8$:      MOVL    28(PACKET), CURRENT_UCB         ; 0522
                50          6B    D0  000A8   MOVL    CURRENT_UCB, R0                         ; 0523
                10          0A    A0  91  000AB   CMPB    10(R0), #16                             ;
                                  04  13  000AF   BEQL    9$                                      ;
```

```
                                FEFF 000B1          BUGW                                                    ; 0524
                               0000* 000B3          .WORD   <BUG$_NOTUCBIRP!4>
                          50    6B  D0 000B5 9$:     MOVL    CURRENT_UCB, R0                                 ; 0526
                0000G     CF 34 A0  D0 000B8        MOVL    52(R0), CURRENT_VCB
                          50 0000G CF D0 000BE      MOVL    CURRENT_VCB, R0                                 ; 0527
                          11   0A A0 91 000C3       CMPB    10(R0), #17
                                   04 13 000C7      BEQL    10$
                                FEFF 000C9          BUGW                                                    ; 0528
                               0000* 000CB          .WORD   <BUG$_NOTVCBUCB!4>
                     0000G     CF   DD 000CD 10$:   PUSHL   IO_CHANNEL                                      ; 0530
                0000V CF           01 FB 000D1      CALLS   #1, GET_CCB
                          55       50 D0 000D6      MOVL    R0, CCB
                          50       6B D0 000D9      MOVL    CURRENT_UCB, R0                                 ; 0531
                          65       50 D0 000DC      MOVL    R0, (CCB)
                          57 1C    A0 D0 000DF      MOVL    28(R0), ORB                                     ; 0532
                          69 18    A6 D0 000E3      MOVL    24(PACKET), CURRENT_WINDOW                      ; 0538
                          02 18    A6 E9 000E7      BLBC    24(PACKET), 11$                                 ; 0539
                                   69 D4 000EB      CLRL    CURRENT_WINDOW                                  ; 0540
                          06 18    A6 93 000ED 11$:     BITB    24(PACKET), #6                              ; 0541
                                   04 13 000F1      BEQL    12$
                                FEFF 000F3          BUGW                                                    ; 0542
                               0000* 000F5          .WORD   <BUG$_BADWCBPT!4>
                          50       69 D0 000F7 12$:     MOVL    CURRENT_WINDOW, R0                          ; 0544
                                   2E 13 000FA      BEQL    15$
                          12   0A  A0 91 000FC      CMPB    10(R0), #18                                     ; 0547
                                   04 13 00100      BEQL    13$
                                FEFF 00102          BUGW                                                    ; 0548
                               0000* 00104          .WORD   <BUG$_NOTWCBIRP!4>
                          50       69 D0 00106 13$:     MOVL    CURRENT_WINDOW, R0                          ; 0550
          04      0B      A0       02 E1 00109      BBC     #2, 11(R0), 14$                                 ; 0551
                                FEFF 0010E          BUGW
                               0000* 00110          .WORD   <BUG$_NOTFCPWCB!4>
                          50       69 D0 00112 14$:     MOVL    CURRENT_WINDOW, R0                          ; 0553
                0000G     CF 18    A0 D0 00115      MOVL    24(R0), PRIMARY_FCB
                          50 0000G CF D0 0011B      MOVL    PRIMARY_FCB, R0                                 ; 0554
                          07   0A  A0 91 00120      CMPB    10(R0), #7
                                   04 13 00124      BEQL    15$
                                FEFF 00126          BUGW                                                    ; 0555
                               0000* 00128          .WORD   <BUG$_NOTFCBWCB!4>
                          16   2A  A6       03 E1 0012A 15$:     BBC     #3, 42(PACKET), 16$                ; 0566
                          50 2C    B6 D0 0012F      MOVL    @44(PACKET), ABD                                ; 0569
                             02    A0 B4 00133      CLRW    2(ABD)                                          ; 0570
                          38 A6 1C A6 D1 00136      CMPL    28(PACKET), 56(PACKET)                          ; 0571
                                   1C 13 0013B      BEQL    17$
                0000G     CF 80    8F 88 0013D      BISB2   #128, CLEANUP_FLAGS                             ; 0572
                                   14 11 00143      BRB     17$                                             ; 0566
          2F    20      A6       06       00 ED 00145 16$:     CMPZV   #0, #6, 32(PACKET), #47             ; 0580
                                   0C 1B 0014B      BLEQU   17$
          38    20      A6       06       00 ED 0014D      CMPZV   #0, #6, 32(PACKET), #56                 ; 0581
                                   04 13 00153      BEQL    17$
                                FEFF 00155          BUGW                                                    ; 0582
                               0000* 00157          .WORD   <BUG$_NOBUFPCKT!4>
                          68   58  B6       30 28 00159 17$:     MOVC3   #48, @88(PACKET), LOCAL_ARB         ; 0589
                          17   03  A8       04 E0 0015E      BBS     #4, LOCAL_ARB+3, 18$                    ; 0590
                          12   03  A8       05 E0 00163      BBS     #5, LOCAL_ARB+3, 18$                    ; 0591
00000000G 9F      3A      A8       10       00 ED 00168      CMPZV   #0, #16, LOCAL_ARB+58, @#EXE$GL_SYSUIC  ; 0592
                                   06 1B 00172      BLEQU   18$
```

```
                 67      38   A8  D1 00174          CMPL    LOCAL_ARB+56, (ORB)                     ; 0593
                             09  12 00178          BNEQ    19$                                     ;
           0000G CF              01  88 0017A 18$:  BISB2   #1, CLEANUP_FLAGS+1                    ; 0596
              03 A8              10  88 0017F       BISB2   #16, LOCAL_ARB+3                        ; 0597
                 50              56  D0 00183 19$:  MOVL    PACKET, R0                              ; 0600
                                 04  00186          RET                                             ; 0602

; Routine Size:  391 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  291     0603  1 GLOBAL ROUTINE GET_CCB (CHANNEL) =
;  292     0604  1
;  293     0605  1 !++
;  294     0606  1 !
;  295     0607  1 ! FUNCTIONAL DESCRIPTION:
;  296     0608  1 !
;  297     0609  1 !       This routine returns the address of the channel control block
;  298     0610  1 !       associated with the given channel.
;  299     0611  1 !
;  300     0612  1 !
;  301     0613  1 ! CALLING SEQUENCE:
;  302     0614  1 !       GET_CCB (ARG1)
;  303     0615  1 !
;  304     0616  1 ! INPUT PARAMETERS:
;  305     0617  1 !       ARG1: channel number
;  306     0618  1 !
;  307     0619  1 ! IMPLICIT INPUTS:
;  308     0620  1 !       NONE
;  309     0621  1 !
;  310     0622  1 ! OUTPUT PARAMETERS:
;  311     0623  1 !       NONE
;  312     0624  1 !
;  313     0625  1 ! IMPLICIT OUTPUTS:
;  314     0626  1 !       NONE
;  315     0627  1 !
;  316     0628  1 ! ROUTINE VALUE:
;  317     0629  1 !       address of CCB
;  318     0630  1 !
;  319     0631  1 ! SIDE EFFECTS:
;  320     0632  1 !       NONE
;  321     0633  1 !
;  322     0634  1 !--
;  323     0635  1
;  324     0636  1
;  325     0637  2 BEGIN
;  326     0638  2
;  327     0639  2 LINKAGE
;  328     0640  2         L_VERIFYCHAN    = JSB (REGISTER = 0) :
;  329     0641  2                           GLOBAL (CCB = 1)
;  330     0642  2                           NOPRESERVE (2, 3, 4, 5);
;  331     0643  2
;  332     0644  2 GLOBAL REGISTER
;  333     0645  2         CCB             = 1 : REF BBLOCK; ! CCB address returned
;  334     0646  2
;  335     0647  2 LOCAL
;  336     0648  2         STATUS,                           ! status of system call
;  337     0649  2         CCB_BASE        : REF BBLOCK;     ! other copy of CCB address, due to
;  338     0650  2                                           ! faulty flow analysis in BLISS
;  339     0651  2
;  340     0652  2 EXTERNAL ROUTINE
;  341     0653  2         IOC$VERIFYCHAN  : L_VERIFYCHAN ADDRESSING_MODE (ABSOLUTE);
;  342     0654  2                                           ! exec routine to find CCB
;  343     0655  2
;  344     0656  2 STATUS = IOC$VERIFYCHAN (.CHANNEL);
;  345     0657  2 CCB_BASE = .CCB;
;  346     0658  2 IF NOT .STATUS
;  347     0659  2 THEN BUG_CHECK (INVCHAN, FATAL, 'Invalid ACP channel number');
```

```
;  348          0660  2
;  349          0661  2 RETURN .CCB_BASE;
;  350          0662  2
;  351          0663  1 END;                                    ! end of routine GET_CCB


                                                        .EXTRN  IOC$VERIFYCHAN, BUG$_INVCHAN

                                            0FFC 00000          .ENTRY  GET_CCB, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,- ; 0603
                                                                        R11                                         ;
                                 50      04 AC  D0 00002        MOVL    CHANNEL, R0                                 ; 0656
                                   00000000G 9F  16 00006        JSB     @#IOC$VERIFYCHAN                            ;
                                 04          50  E8 0000C        BLBS    STATUS, 1$                                  ; 0658
                                               FEFF 0000F        BUGW                                                ; 0659
                                              0000* 00011        .WORD   <BUG$_INVCHAN!4>                            ;
                                 50           51  D0 00013 1$:   MOVL    CCB_BASE, R0                                ; 0661
                                                  04 00016        RET                                                 ; 0663

; Routine Size:  23 bytes,    Routine Base:  $CODE$ + 0187

;  352          0664  1
;  353          0665  1 END
;  354          0666  0 ELUDOM


;                                   PSECT SUMMARY
;
;      Name                     Bytes                          Attributes
;  $CODE$                         414  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                                 Library Statistics
;                                          --------- Symbols ---------      Pages       Processing
;      File                                Total     Loaded   Percent      Mapped      Time
;  _$255$DUA28:[SYSLIB]LIB.L32;1           18619         43         0        1000        00:01.9


;                                   COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:GETREQ/OBJ=OBJ$:GETREQ MSRC$:GETREQ/UPDATE=(ENH$:GETREQ)
```

```
; Size:          414 code + 0 data bytes
; Run Time:        00:12.2
; Elapsed Time:    00:32.8
; Lines/CPU Min:     3264
; Lexemes/CPU-Min: 17343
; Memory Used:  161 pages
; Compilation Complete
```

EXTFCB.LIS

DELFIL.LIS

DIRGET.LIS

EXTIDX.LIS

IODONE.LIS

LOCKDN.LIS

ENTER.LIS

GETREQ.LIS

GETTIM.LIS

DISPAT.LIS

INIFCP.LIS

DIRFCB.LIS

EXTHDR.LIS

DIRSCN.LIS

LOGDEL.LIS

LOCKDB.LIS

FIND.LIS

GETFIB.LIS

DIRACC.LIS

INIFCB.LIS

EXTDIR.LIS

EXTEND.LIS