L 5

```
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS  LLL                 OOOOOOOOO        AAAAAAAAA
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS  LLL                 OOOOOOOOO        AAAAAAAAA
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS  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
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS     LLLLLLLLLLLLLLL     OOOOOOOOO     AAA         AAA
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS     LLLLLLLLLLLLLLL     OOOOOOOOO     AAA         AAA
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS     LLLLLLLLLLLLLLL     OOOOOOOOO     AAA         AAA
```

FILEID**CONMAN

```
  CCCCCCCC    OOOOOO    NN      NN  MM      MM    AAAAAA    NN      NN
  CCCCCCCC    OOOOOO    NN      NN  MM      MM    AAAAAA    NN      NN
CC          OO      OO  NN      NN  MMMM  MMMM  AA      AA  NN      NN
CC          OO      OO  NN      NN  MMMM  MMMM  AA      AA  NN      NN
CC          OO      OO  NNNN    NN  MM  MM  MM  AA      AA  NNNN    NN
CC          OO      OO  NNNN    NN  MM  MM  MM  AA      AA  NNNN    NN
CC          OO      OO  NN  NN  NN  MM      MM  AA      AA  NN  NN  NN
CC          OO      OO  NN  NN  NN  MM      MM  AA      AA  NN  NN  NN
CC          OO      OO  NN    NNNN  MM      MM  AAAAAAAAAA  NN    NNNN
CC          OO      OO  NN    NNNN  MM      MM  AAAAAAAAAA  NN    NNNN
CC          OO      OO  NN      NN  MM      MM  AA      AA  NN      NN      ....
CC          OO      OO  NN      NN  MM      MM  AA      AA  NN      NN      ....
  CCCCCCCC    OOOOOO    NN      NN  MM      MM  AA      AA  NN      NN      ....
  CCCCCCCC    OOOOOO    NN      NN  MM      MM  AA      AA  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  CONMAN - Cluster Configuration Manager
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: EXECUTIVE, CLUSTER MANAGEMENT
0000  ;
0000  ; ABSTRACT:
0000  ;       This module creates and manages the cluster configuration.  It is
0000  ;       responsible for forming a cluster, adding nodes to a cluster, and
0000  ;       reconfiguring a cluster following a failure.
0000  ;
0000  ; ENVIRONMENT: VAX/VMS
0000  ;
0000  ; AUTHOR: David W. Thiel,        CREATION DATE: 4-Apr-1983
0000  ;
0000  ; MODIFIED BY:
0000  ;
0000  ;       V03-022 DWT0225         David W. Thiel          11-Jul-1984
0000  ;               Change call to temporary name EXE$MNTVERSP2 to
0000  ;               real name EXE$CLUTRANIO.
0000  ;
0000  ;       V03-021 DWT0219         David W. Thiel          08-May-1984
0000  ;               Add synchronization between I/O and lock manager
0000  ;               clusters.
0000  ;
0000  ;       V03-020 DWT0213         David W. Thiel          17-Apr-1984
0000  ;               Always fork at least once when locking nodes to
0000  ;               begin a state transition.  Correct logic for
0000  ;               locking nodes when memory allocation fails.
0000  ;
0000  ;       V03-019 DWT0206         David W. Thiel          08-Apr-1984
0000  ;               Correct branch in CNX$RCVD_RECONFIG.
0000  ;               Tighten up timing on cluster formation to avoid
```

```
0000 ;                        forming a paritioned cluster.  Add CNX$POWER_FAIL
0000 ;                        entry point to receive control on power recovery.
0000 ;                        Use CLUBFKB$V_FKB_BUSY instead of CLUB$V_FKB_BUSY.
0000 ;                        Fix method of comparing quorum disk names from
0000 ;                        various nodes.
0000 ;
0000 ;               V03-018 DWT0201          David W. Thiel          24-Mar-1984
0000 ;                        Remove temporary code defining new symbols in the CLUB.
0000 ;
0000 ;               V03-017 DWT0194          David W. Thiel          23-Mar-1984
0000 ;                        Change interface to disk quorum module to eliminate
0000 ;                        window of ambiguity.  Rename CLUB$V_QF_SKIP_READ to
0000 ;                        CLUB$V_QF_FAILED_NODE.  Update CSB$Q_REFTIME when a node
0000 ;                        is added to or removed from the cluster.  Make sending
0000 ;                        phase 1 messages a subroutine.  Make addition/deletion
0000 ;                        of quorum disk from cluster a full two-phase transition.
0000 ;                        Support requests for quorum change.  Remove propagation
0000 ;                        of some bits to all nodes of cluster.
0000 ;
0000 ;               V03-016 DWT0187          David W. Thiel          5-Mar-1984
0000 ;                        Correct and improve random number time-outs.
0000 ;
0000 ;               V03-015 DWT0181          David W. Thiel          27-Feb-1984
0000 ;                        Correct MULL3 to be MULL2.
0000 ;
0000 ;               V03-014 DWT0173          David W. Thiel          27-Feb-1984
0000 ;                        Start own reconfiguration attempts after some other
0000 ;                        node tries and fails.  Raise interval to 0...20 seconds
0000 ;                        when requesting cluster membership.  Correct use of
0000 ;                        CLUB$W_QDVOTES to always minimize value and to propagate
0000 ;                        when forming or adding nodes.  Add dummy entry
0000 ;                        CNX$SHUTDOWN, update dummy CNX$CHANGE_QUORUM.
0000 ;
0000 ;               V03-013 DWT0161          David W. Thiel          03-Feb-1984
0000 ;                        Modify use of CSB$B_NODEMAP to represent current
0000 ;                        best estimate of global connection status.  Bit 0
0000 ;                        is used for the quorum disk "connection".  This field
0000 ;                        is maintained for use by the optimal sub-cluster
0000 ;                        calculation.  Do optimal calculation on cluster
0000 ;                        reconfigurations and cluster formation.  Always
0000 ;                        copy CLUB$V_ACTIVE bit to CSB$V_ACTIVE bit in
0000 ;                        CNX$DISK_CHANGE.  Correct blown register.
0000 ;
0000 ;               V03-012 DWT0147          David W. Thiel          19-Dec-1983
0000 ;                        Add support for lock manager directory system vector.
0000 ;                        Remove old lock manager directory system logic.
0000 ;                        Add CLUB$W_MEMSEQ (cluster membership state sequence
0000 ;                        number) and support for maintaining same.  Add use of
0000 ;                        CLMPRO$L_FMERIT field and its propagation with phase
0000 ;                        1 messages.  Use common routine CNX$QUORUM_CALC for
0000 ;                        all quorum calculations.  Add support for a variable
0000 ;                        number of votes for the quorum disk.  Remove filling
0000 ;                        in several unused fields in the CLMSTS message.
0000 ;                        Invoke fail-in table on all cluster membership
0000 ;                        transitions.  Change name of CNX$BEGIN_FAILIN and
0000 ;                        CNX$BEGIN_FAILOVER to CNX$MEMBERSHIP_CHANGE.
0000 ;                        Use routines in new CONSUBS module.  Update TOPOLOGY
```

```
0000    ;               message logic.
0000    ;
0000    ;       V03-011 DWT0141         David W. Thiel          04-Nov-1983
0000    ;               Correct decision of who is to bugcheck when two
0000    ;               clusters meet.
0000    ;
0000    ;       V03-010 DWT0136         David W. Thiel          04-Oct-1983
0000    ;               Correct validation of transaction number in received
0000    ;               message to compare against the current transaction
0000    ;               rather than the maximum transaction number seen.
0000    ;               Use CLUB$V_LOST_CNX to mark situations where there
0000    ;               is a broken connection to a cluster member.
0000    ;
0000    ;       V03-009 DWT0132         David W. Thiel          23-Sep-1983
0000    ;               Correct misuse of register on an error path in
0000    ;               CNX$RCVD_DESC.  Improve state change messages to
0000    ;               be less repetitious.
0000    ;
0000    ;       V03-008 DWT0128         David W. Thiel          30-Aug-1983
0000    ;               Be less interested in received status when CLUB$V_INIT
0000    ;               flag has not yet been set -- avoid attempting to join
0000    ;               a cluster in this situation.
0000    ;               Enable writing quorum file when node is added to an
0000    ;               existing cluster.
0000    ;
0000    ;       V03-007 DWT0118         David W. Thiel          22-Aug-1983
0000    ;               Add quorum disk support.  Change CONFIG_CHANGE to
0000    ;               CNX$CONFIG_CHANGE.  Add logic to recover from a
0000    ;               failure while a cluster state transition is in
0000    ;               progress.  Add routines to process incoming QUORUM
0000    ;               and TOPOLOGY messages.
0000    ;
0000    ;       V03-006 DWT0116         David W. Thiel          1-Aug-1983
0000    ;               Change INIT_STD_RESP to CNX$INIT_STD_RESP and make
0000    ;               symbol gobal.  Change PROCESS_RESPONSE to
0000    ;               CNX$PROCESS_REPONSE and make symbol global.
0000    ;               Improve use of CLUB$L_TQE field.
0000    ;
0000    ;       V03-005 DWT0111         David W. Thiel          28-Jul-1983
0000    ;               Automatically raise quorum to be greater than half
0000    ;               of the sum of available votes.
0000    ;               Use real TQE for random timeouts.
0000    ;               Add CNX$CHANGE_QUORUM entry point.
0000    ;               Correct handling of status bits received from a
0000    ;               remote node.
0000    ;               Hang if no quorum using CNX$CHECK_QUORUM routine.
0000    ;
0000    ;       V03-004 DWT0109         David W. Thiel          25-Jun-1983
0000    ;               Select appropriate nodes when join message is received.
0000    ;               Defer setting directory system in joining node
0000    ;               until phase 2 message is received.
0000    ;
0000    ;       V03-003 DWT0108         David W. Thiel          23-Jun-1983
0000    ;               Fix broken BLBC instruction.  Fix logic for looking
0000    ;               up a node described in a message from a remote system.
0000    ;
0000    ;       V03-002 DWT0106         David W. Thiel          23-Jun-1983
```

```
0000    ;               Correct search for CSB when node description received.
0000    ;               Correct propagation of directory node when a node
0000    ;               joins the cluster.
0000    ;               Deallocate RSPID after use in DESCRIBE_NODE.
0000    ;               When a connection breaks, disable locks and forbid
0000    ;               additions of nodes to the cluster.
0000    ;               Correct discrepancy in protocol for reqeusting membership
0000    ;               in a cluster.
0000    ;
0000    ;       V03-001 DWT0103         David W. Thiel          27-May-1983
0000    ;               Fix blown register.
0000    ;
0000    ;--
0000
```

```
    0000          .SBTTL  DECLARATIONS
    0000  ;
    0000  ; INCLUDE FILES:
    0000  ;
    0000          $CDRPDEF                        ; CDRP offsets
    0000          $CDTDEF                         ; CDT Offsets
    0000          $CLMDRSDEF                      ; Cluster disconnect/reject codes
    0000          $CLSMSGDEF                      ; Cluster message definitions
    0000          $CLUBDEF                        ; CLUster Block offsets
    0000          $CSBDEF                         ; CSB Offsets
    0000          $DYNDEF                         ; Data structure type codes
    0000          $FKBDEF                         ; ForK Block offsets
    0000          $IPLDEF                         ; IPL definitions
    0000          $PCBDEF                         ; Process Control Block offsets
    0000          $PDTDEF                         ; Port Definition Table offsets
    0000          $SBDEF                          ; System Block offsets
    0000          $SSDEF                          ; Status code definitions
    0000          $TQEDEF                         ; TQE offsets
    0000
    0000  ;
    0000  ; OWN STORAGE:
    0000  ;
    0000
00000000          .PSECT  $$$040,LONG             ; R/W Data PSECT
    0000
    0000  ;***********************************************************************
    0000  ;
    0000  ; NOTE: The following assumptions are in effect for this entire module.
    0000  ;
    0000  ;***********************************************************************
    0000
    0000          ASSUME  IPL$_SYNCH  EQ  IPL$_SCS
    0000          ASSUME  IPL$_SYNCH  EQ  IPL$_TIMER
    0000
00000000          .PSECT  $$$100,LONG
    0000
    0000          .DEFAULT        DISPLACEMENT,WORD
```

```
                              0000          .SBTTL  CNX$CON_INIT - Initialization Call
                              0000
                              0000    ;++
                              0000    ;
                              0000    ; FUNCTIONAL DESCRIPTION:
                              0000    ;
                              0000    ;       This routine is called at system initialization to start up the
                              0000    ;       configuration manager.  This call must be made AFTER the CLUB
                              0000    ;       is created and initialized.  IPL is 31.
                              0000    ;
                              0000    ; CALLING SEQUENCE:
                              0000    ;
                              0000    ;       JSB     CNX$CON_INIT
                              0000    ;       IPL is 31
                              0000    ;
                              0000    ; INPUT PARAMETERS:
                              0000    ;
                              0000    ;       NONE
                              0000    ;
                              0000    ; OUTPUT PARAMETERS:
                              0000    ;
                              0000    ;       NONE
                              0000    ;
                              0000    ; COMPLETION CODES:
                              0000    ;
                              0000    ;       R0 contains status.
                              0000    ;
                              0000    ; SIDE EFFECTS:
                              0000    ;
                              0000    ;       R1 is destroyed.
                              0000    ;
                              0000    ;--
                              0000
                          00000000          .PSECT  $$$002,LONG                 ; Initialization Psect
                              0000
                              0000    CNX$CON_INIT::
                  3C    BB    0000          PUSHR   #^M<R2,R3,R4,R5>            ; Save registers
   54    00000000'GF    D0    0002          MOVL    G^CLU$GL_CLUB,R4            ; Address of CLUB
     00 1C A4     12    E2    0009          BBSS    #CLUB$V_NO_FORM, -          ; Set bit inhibiting cluster formation
                              000E                  CLUB$L_FLAGS(R4),10$
        53    10 A4     D0    000E    10$:  MOVL    CLUB$L_LOCAL_CSB(R4),R3     ; Address of local CSB
 60 A3    02000008 8F   C8    0012          BISL2   #<CSB$M_STATUS_RCVD ! -     ; Mark status received
                              001A                    CSB$M_QF_SAME>, -         ; Mark quorum files matching
                              001A                  CSB$L_STATUS(R3)            ;   for local CSB
           51     18    3C    001A          MOVZWL  #FKB$K_LENGTH,R1            ; Fork block length
               FFE0'    30    001D          BSBW    CNX$ALLOZMEM                ; Allocate and zero a block of memory
              11 50     E9    0020          BLBC    R0,20$                      ; Branch on error
              55 52     D0    0023          MOVL    R2,R5
        0A A5     08    90    0026          MOVB    #DYN$C_FRK, -               ; Block type
                              002A                  FKB$B_TYPE(R5)
        0B A5     08    90    002A          MOVB    #IPL$_SCS, -                ; Fork IPL
                              002E                  FKB$B_FIPL(R5)
               FFCF'    30    002E          BSBW    INIT_CONTINUE               ; Put synchronous return on stack
           50     01    D0    0031          MOVL    S^#SS$_NORMAL,R0            ; Set success status
                  3C    BA    0034    20$:  POPR    #^M<R2,R3,R4,R5>            ; Restore registers
                        05    0036          RSB
                              0037
```

```
                    00000000        .PSECT  $$$100,LONG
                        0000
                        0000  INIT_CONTINUE:
      00000000'GF   16  0000        JSB     G^EXE$FORK              ; Fork
                        0006  ;
                        0006  ; Get here soon after IPL is lowered to SCS (8) at system initialization
                        0006  ;
             FFF7'  30  0006        BSBW    CNX$RANDOM_INIT         ; Initialize random number generator
                        0009  ;
                        0009  ; Wait for SYSINIT to start up and declare us ready for think about forming a cluste
                        0009  ;
              0269  30  0009  10$:  BSBW    CNX$SEND_ALL_STATUS     ; Send status as necessary
                        000C        FORK_WAIT                       ; Wait a second
   F2 1C A4   13    E1  0012        BBC     #CLUB$V_INIT, -         ; Branch if SYSINIT has not yet triggered us
                        0017                CLUB$L_FLAGS(R4),10$
                30  BB  0017        PUSHR   #^M<R4,R5>              ; Save registers
              0539  30  0019        BSBW    SCAN                    ; Look for work to do
                30  BA  001C        POPR    #^M<R4,R5>              ; Restore registers
                        001E  ;
                        001E  ; We must now delay an appropriate time before forming a cluster in order to allow
                        001E  ; the discovery of other nodes that are already cluster members before attempting
                        001E  ; to form our own cluster.  This time is the larger of:
                        001E  ;       a) SCS process poller interval + largest PDT polling sweep interval
                        001E  ;       b) Time to find quorum disk device and acquire a "connection" to the
                        001E  ;               quorum disk
                        001E  ; In fact, we ignore the latter at this time.
                        001E  ; We do the wait in two steps.  First we wait the process poller interval because
                        001E  ; this is known.  Then we wait a time which is the maximum of the polling sweep
                        001E  ; intervals of all ports that have been found during the earlier interval.
                        001E  ;
 53   00000000'GF   3C  001E        MOVZWL  G^SCS$GW_PRCPOLINT,R3   ; Process polling interval
                        0025  20$:  FORK_WAIT                       ; Wait a second
   32 1C A4   00    E0  002B        BBS     #CLUB$V_CLUSTER, -      ; Branch if a cluster has been formed
                        0030                CLUB$L_FLAGS(R4),60$
             F2 53  F4  0030        SOBGEQ  R3,20$                  ; Wait for interval to complete
                        0033        ;
                        0033        ; By now, all SCS ports should have been found.
                        0033        ; Scan list of all PDT's computing the largest value of PDT$L_POLLSWEEP that
                        0033        ; can be found.  This is the interval that we should now wait before
                        0033        ; forming a cluster, i.e. it is the interval that may elapse before a
                        0033        ; connection may be made to an existing cluster node.
                        0033        ;
                53  D4  0033        CLRL    R3                      ; Accumulator for larger PDT$L_POLLSWEEP
 50   00000000'GF   DE  0035        MOVAL   G^SCS$GL_PDT,R0         ; Get address of PDT listhead
           50   60  D0  003C  30$:  MOVL    (R0),R0                 ; Link to next PDT
                0E  13  003F        BEQL    50$                     ; Branch when all done
     53   00D8  C0  D1  0041        CMPL    PDT$L_POLLSWEEP(R0),R3  ; Is this value larger?
                05  1B  0046        BLEQU   40$                     ; Branch if no
     53   00D8  C0  D0  0048        MOVL    PDT$L_POLLSWEEP(R0),R3  ; Update largest seen yet
                ED  11  004D  40$:  BRB     30$                     ; Continue iteration
                        004F
                        004F  50$:  FORK_WAIT                       ; Wait a second
   08 1C A4   00    E0  0055        BBS     #CLUB$V_CLUSTER, -      ; Branch if a cluster has been formed
                        005A                CLUB$L_FLAGS(R4),60$
             F2 53  F4  005A        SOBGEQ  R3,50$                  ; Wait for interval to complete
   00 1C A4   12    E5  005D        BBCC    #CLUB$V_NO_FORM, -      ; Clear formation inhibit
                        0062                CLUB$L_FLAGS(R4),60$
```

```
      50    55  D0 0062   339 60$:  MOVL    R5,R0                   ; Fork block address
 00000000'GF    16 0065   340       JSB     G^EXE$DEANONPAGED       ; Deallocate fork block
        04E7    31 006B   341       BRW     SCAN                    ; Look for work to do
```

```
                        006E            .SBTTL  CNX$POWER_FAIL - Power Fail Recovery Entry
                        006E
                        006E    ;++
                        006E    ;
                        006E    ; FUNCTIONAL DESCRIPTION:
                        006E    ;
                        006E    ;       This routine is called during recovery from a power failre.
                        006E    ;       IPL is 31.
                        006E    ;
                        006E    ; CALLING SEQUENCE:
                        006E    ;
                        006E    ;       JSB     CNX$POWER_FAIL
                        006E    ;       IPL is 31
                        006E    ;
                        006E    ; INPUT PARAMETERS:
                        006E    ;
                        006E    ;       NONE
                        006E    ;
                        006E    ; OUTPUT PARAMETERS:
                        006E    ;
                        006E    ;       NONE
                        006E    ;
                        006E    ; COMPLETION CODES:
                        006E    ;
                        006E    ;       NONE
                        006E    ;
                        006E    ; SIDE EFFECTS:
                        006E    ;
                        006E    ;       All registers are preserved
                        006E    ;
                        006E    ;--
                        006E
                        006E    CNX$POWER_FAIL::
             38    BB   006E            PUSHR   #^M<R3,R4,R5>             ; Save registers
54   00000000'GF   D0   0070            MOVL    G^CLU$GL_CLUB,R4          ; Address of CLUB
    55     018C C4 9E   0077            MOVAB   CLUB$B_CLUBPWF(R4),R5     ; Fork block address
   02 18 A5     00 E2   007C            BBSS    #CLUBPWF$V_BUSY, -        ; Branch if block is in use
                        0081                    CLUBPWF$L_STATUS(R5),10$ ;   by a previous power recovery
             03    10   0081            BSBB    20$                       ; Save synchronous return address
             38    BA   0083    10$:    POPR    #^M<R3,R4,R5>             ; Restore registers
                   05   0085            RSB                               ; Return to power fail notification code
                        0086
     00000000'GF   16   0086    20$:    JSB     G^EXE$FORK                ; Fork to IPL SYNCH, SCS
                        008C    ;
                        008C    ; Continue here at IPL 8
                        008C    ;
   20 18 A5     00 E5   008C            BBCC    #CLUBPWF$V_BUSY, -        ; Clear busy bit, branch on error
                        0091                    CLUBPWF$L_STATUS(R5),60$
   00 1C A4     1E E5   0091            BBCC    #CLUB$V_QF_DYNVOTE, -     ; Clear bit counting disk in dynamic quorum
                        0096                    CLUB$L_FLAGS(R4),30$
   00 1C A4     18 E2   0096    30$:    BBSS    #CLUB$V_QF_FAILED_NODE, - ; Trigger quorum disk manager to reacquire
                        009B                    CLUB$L_FLAGS(R4),40$
      52     1C A4 D0   009B    40$:    MOVL    CLUB$L_FLAGS(R4),R2       ; Save flags
           FF5E'   30   009F            BSBW    CNX$CHECK_QUORUM          ; Check for out of quorum condition
      52     1C A4 CA   00A2            BICL2   CLUB$L_FLAGS(R4),R2       ; Set flag in R2 if clear now and set earlie
   06 52        1C E0   00A6            BBS     #CLUB$V_QUORUM,R2,50$     ; Branch if quorum was just lost and mntver
                        00AA                                              ;   triggered
```

```
00000000'GF  16  00AA   400         JSB     G^EXE$CLUTRANIO          ; Throw disks into mount verification to ens
                 00B0   401                                          ;    all disk I/O is blocked, pending resolut
                 00B0   402                                          ;    the status of the cluster
             05  00B0   403 50$:    RSB
                 00B1   404
                 00B1   405 60$:    BUG_CHECK       CNXMGRERR,FATAL ; Multiple use of fork block
                 00B5   406
```

```
                        00B5          .SBTTL  CNX$CON_NEWSYS - Connected to New System
                        00B5
                        00B5  ;++
                        00B5  ;
                        00B5  ; FUNCTIONAL DESCRIPTION:
                        00B5  ;
                        00B5  ;       This routine is called when a connection to a new system is
                        00B5  ;       made.  It will exchange status with the new system and consider
                        00B5  ;       forming a cluster, joining the cluster of which the new system is
                        00B5  ;       a member, or wait for the new system to join the cluster of which
                        00B5  ;       this system is a member.
                        00B5  ;
                        00B5  ; CALLING SEQUENCE:
                        00B5  ;
                        00B5  ;       JSB     CNX$CON_NEWSYS
                        00B5  ;       IPL is IPL$_SCS
                        00B5  ;
                        00B5  ; INPUT PARAMETERS:
                        00B5  ;
                        00B5  ;       R5: CSB of new system
                        00B5  ;
                        00B5  ; OUTPUT PARAMETERS:
                        00B5  ;
                        00B5  ;       NONE
                        00B5  ;
                        00B5  ; COMPLETION CODES:
                        00B5  ;
                        00B5  ;       NONE
                        00B5  ;
                        00B5  ; SIDE EFFECTS:
                        00B5  ;
                        00B5  ;       R0 and R1 are destroyed
                        00B5  ;
                        00B5  ;--
                        00B5
                        00B5  CNX$CON_NEWSYS::
            3C    BB    00B5          PUSHR   #^M<R2,R3,R4,R5>          ; Save registers
        53  55    D0    00B7          MOVL    R5,R3                     ; Put CSB address in R3
  00 60 A3  1A    E3    00BA          BBCS    #CSB$V_SEND_STATUS, -     ; Need to send status
                        00BF                  CSB$L_STATUS(R3),10$
          03F4  30      00BF  10$:    BSBW    START_TIMEOUT             ; Start CLUB-based timer
            3C    BA    00C2          POPR    #^M<R2,R3,R4,R5>          ; Restore registers
                  05    00C4          RSB                               ; Return
```

```
                                00C5          .SBTTL  CNX$CON_BREAK - Connection has broken
                                00C5
                                00C5  ;++
                                00C5  ;
                                00C5  ; FUNCTIONAL DESCRIPTION:
                                00C5  ;
                                00C5  ;       This routine is called when a connection to a system has broken and
                                00C5  ;       attempts to reclose the connection have failed.
                                00C5  ;
                                00C5  ; CALLING SEQUENCE:
                                00C5  ;
                                00C5  ;       JSB     CNX$CON_BREAK
                                00C5  ;       IPL is IPL$_SCS
                                00C5  ;
                                00C5  ; INPUT PARAMETERS:
                                00C5  ;
                                00C5  ;       R5: CSB of system to which the connection is lost
                                00C5  ;
                                00C5  ; OUTPUT PARAMETERS:
                                00C5  ;
                                00C5  ;       NONE
                                00C5  ;
                                00C5  ; COMPLETION CODES:
                                00C5  ;
                                00C5  ;       NONE
                                00C5  ;
                                00C5  ; SIDE EFFECTS:
                                00C5  ;
                                00C5  ;       R0 and R1 are destroyed
                                00C5  ;
                                00C5  ;--
                                00C5
                                00C5  CNX$CON_BREAK::
                      3C  BB    00C5          PUSHR   #^M<R2,R3,R4,R5>              ; Save registers
                  53  55  D0    00C7          MOVL    R5,R3                         ; Put CSB address in R3
              54  64 A3  D0    00CA          MOVL    CSB$L_CLUB(R3),R4             ; Address of CLUB
          55   010C C4  9E    00CE          MOVAB   CLUB$B_CLUFCB(R4),R5          ; Address of CLUFCB
                                00D3
         22 60 A3   01  E1    00D3          BBC     #CSB$V_MEMBER, -              ; Branch if lost node is
                                00D8                  CSB$L_STATUS(R3),10$         ;   not a cluster member
   1C A4    00800000 8F  C8    00D8          BISL2   #CLUB$M_LOST_CNX, -           ; Mark connection to member lost
                                00E0                  CLUB$L_FLAGS(R4)
       00000000'GF   01  8E    00E0          MNEGB   #1,G^LCK$GB_STALLREQS         ; Stall lock requests
            50    4C A3  3C    00E7          MOVZWL  CSB$W_CSID_IDX(R3),R0         ; CSID index value
         00 28 A5   50  E5    00EB          BBCC    R0, -                         ; Clear participation bit in
                                00F0                  CLUFCB$B_NODEMAP(R5),5$      ;   failover table for lost node
            51    10 A4  D0    00F0  5$:     MOVL    CLUB$L_LOCAL_CSB(R4),R1       ; Local node CSB address
       00 008C C1   50  E5    00F4          BBCC    R0,CSB$B_NODEMAP(R1),10$      ; Mark loss of connectivity from local node
                                00FA  10$:
         32 1C A4   1D  E1    00FA          BBC     #CLUB$V_TRANSITION, -         ; Branch if no state change in
                                00FF                  CLUB$L_FLAGS(R4), 60$        ;   progress
      10 A4     5C A4  D1    00FF          CMPL    CLUB$L_COORD(R4), -           ; Is this node the coordinator
                                0104                  CLUB$L_LOCAL_CSB(R4)
                      1F  13    0104          BEQL    40$                           ; Branch if this node is the coordinator
                                0106  ;
                                0106  ; Here if transition is in progress and this node is not the coordinator
                                0106  ;
```

```
     53    5C A4  D1  0106          CMPL    CLUB$L_COORD(R4),R3        ; Is connection to coordinator lost?
           28     12  010A          BNEQ    70$                        ; Branch if not coordinator connection
                      010C  ;
                      010C  ; The connection to the coordinator has been lost and this node is in the midst of
                      010C  ; a cluster state transition.  If a Phase 1 message has not yet been received, it is
                      010C  ; safe to back out the transition because we know that noone can have seen a Phase 2
                      010C  ; message.
                      010C  ;
  40 8F    59 A4  91  010C          CMPB    CLUB$B_CUR_PHASE(R4), -   ; Has Phase 1 been seen yet?
                      0111                  #CLMCNX$K_PH1
           0D     1F  0111          BLSSU   30$                        ; Branch if no Phase 1 yet
                      0113  ;
                      0113  ; This node has received a Phase 1 message.  If this node is not a member of a
                      0113  ; cluster, then bugcheck -- we can't cleanly resolve what might be happening.
                      0113  ;
  04 1C A4    00  E1  0113          BBC     #CLUB$V_CLUSTER, -         ; Branch if not a cluster member and
                      0118                  CLUB$L_FLAGS(R4),20$       ;   resolve the situation the easy way
           1D     10  0118          BSBB    ANALYZE_PHASE              ; Make complex decision about what to do
           18     11  011A          BRB     70$
                      011C  ;
                      011C  ; This node is not a member of a cluster, is in Phase 1 of a cluster state transitio
                      011C  ; and has lost the connection to the transition coordinator.  It is not trivial
                      011C  ; to resolve this situation.  Take the easy way out.
                      011C  ;
                      011C  20$:    BUG_CHECK       CLUEXIT,FATAL      ; Can't handle this failure case
                      0120  ;
         1123     30  0120  30$:    BSBW    UNLOCK_ALL                 ; Unlock things
           0F     11  0123          BRB     70$
                      0125
                      0125  40$:
                      0125  ;
                      0125  ; Here if transition is in progress and this node is the coordinator
                      0125  ;
  60 8F    59 A4  91  0125          CMPB    CLUB$B_CUR_PHASE(R4), -   ; Has Phase 2 been sent yet?
                      012A                  #CLMCNX$K_PH2
           05     1E  012A          BGEQU   50$                        ; Branch if phase 2 sent
  00 1C A4    11  E3  012C          BBCS    #CLUB$V_UNLOCK, -          ; No phase 2 sent -- request
                      0131                  CLUB$L_FLAGS(R4), 50$      ;   unlock
                      0131  50$:
                      0131  ;
                      0131  ; Common exit routine
                      0131  ;
         0421     30  0131  60$:    BSBW    SCAN                       ; Start CLUB-based timer
           3C     BA  0134  70$:    POPR    #^M<R2,R3,R4,R5>           ; Restore registers
                  05  0136          RSB                                ; Return
```

```
0137            .SBTTL  ANALYZE_PHASE - Determine Transition Phase and Finish Transition
0137
0137    ;++
0137    ;
0137    ; FUNCTIONAL DESCRIPTION:
0137    ;
0137    ;       This routine is called when all of the following are true:
0137    ;       1)      The local node is a member of a cluster
0137    ;       2)      A cluster state transition is in progress
0137    ;       3)      The connection to the coordinator node has broken
0137    ;       4)      A Phase 1 message has been received
0137    ;
0137    ;       This routine must determine whether to commit or backout the transition.
0137    ;
0137    ; CALLING SEQUENCE:
0137    ;
0137    ;       JSB     ANALYZE_PHASE
0137    ;       IPL is IPL$_SCS
0137    ;       The caller is returned to synchronously, but the function performed
0137    ;       continues asynchronously and no notification is given to the caller
0137    ;       upon completion.
0137    ;
0137    ; INPUT PARAMETERS:
0137    ;
0137    ;       R3: CSB of system to which the connection is lost
0137    ;       R4: Address of CLUB
0137    ;
0137    ; OUTPUT PARAMETERS:
0137    ;
0137    ;       NONE
0137    ;
0137    ; COMPLETION CODES:
0137    ;
0137    ;       NONE
0137    ;
0137    ; SIDE EFFECTS:
0137    ;
0137    ;       R0 through R5 are destroyed
0137    ;
0137    ; ALGORITHM:
0137    ;
0137    ;       The strategy here is to poll all remaining cluster members and
0137    ;       see whether they can resolve out dilemma.
0137    ;       If another member has:
0137    ;               Not yet seen Phase 1 ==> Undo the transition in progress
0137    ;               Already seen Phase 2 ==> Commit the transition in progress
0137    ;               Seen Phase 1 and has a coordinator connection ==> Restart poll
0137    ;               Seen Phase 1 and has no coordinator connection ==> Continue poll
0137    ;       If this completes without yield a definitive (Undo or Commit) result,
0137    ;       poll all nodes again.  If another member has:
0137    ;               Not yet seen Phase 1 ==> Undo the transition in progress
0137    ;               Already seen Phase 2 ==> Commit the transition in progress
0137    ;               Phase 1 and has a coordinator connection ==> inconsistent with above
0137    ;               Phase 1 and has no coordinator connection ==>
0137    ;                       Obtain promise from node that it will never commit this
0137    ;                       transition
0137    ;                       Continue poll
```

```
                               0137     ;       If this completes without yielding a definitive (Undo or Commit) result,
                               0137     ;       do not commit the transition.
                               0137     ;
                               0137     ;--
                               0137
                               0137     ANALYZE_PHASE:
        1C A4   00100000 8F  CA 0137            BICL2   #CLUB$M_BACKOUT, -        ; Clear backout inhibit bit
                               013F                    CLUB$L_FLAGS(R4)
        1C A4   00000100 8F  CA 013F     10$:   BICL2   #CLUB$M_STS_PPHASE, -     ; Clear polling phase flag (Phase A)
                               0147                    CLUB$L_FLAGS(R4)
        1C A4   00001E00 8F  CA 0147     20$:   BICL2   #< -                      ; Clear poll result bits
                               014F                    CLUB$M_STS_PH0 ! -        ;    phase 0 seen
                               014F                    CLUB$M_STS_PH1 ! -        ;    phase 1 seen (good coordinator connectio
                               014F                    CLUB$M_STS_PH1B ! -       ;    phase 1 seen (broken coordinator connect
                               014F                    CLUB$M_STS_PH2>, -        ;    phase 2 seen
                               014F                    CLUB$L_FLAGS(R4)
                   5A A4     B4 014F            CLRW    CLUB$W_MSGCNT(R4)         ; Initialize waiting count
                   FEAB'     30 0152            BSBW    CNX$SCAN_CSBS             ; Iterate over all CSBs
                   4D 50     E9 0155            BLBC    R0,70$                    ; Branch when done
           18 60 A3   01     E1 0158            BBC     #CSB$V_MEMBER, -          ; Branch if not a member
                               015D                    CSB$L_STATUS(R3),30$
           13 60 A3   00     E0 015D            BBS     #CSB$V_LONG_BREAK, -      ; Branch if permanent break
                               0162                    CSB$L_STATUS(R3),30$      ;    in connection
           0E 60 A3   18     E0 0162            BBS     #CSB$V_LOCAL, -           ; Branch if local CSB
                               0167                    CSB$L_STATUS(R3),30$
                   FE96'     30 0167            BSBW    CNX$ALLOC_WARMCDRP_CSB
                   09 50     E9 016A            BLBC    R0,40$                    ; No memory available
                   5A A4     B6 016D            INCW    CLUB$W_MSGCNT(R4)         ; Include in wait count
                      07     10 0170            BSBB    50$                       ; Interogate node
                   FE8B'     31 0172            BRW     CNX$SCAN_CSBS_FORK        ; Release control mementarily
                               0175
                             05 0175     30$:   RSB                               ; Return and continue scan
                               0176
                   FE87'     31 0176     40$:   BRW     CNX$SCAN_CSBS_RETRY       ; Delay and resume scan
                               0179     ;
                               0179     ; Send status inquiry to one node
                               0179     ;
                 50   0E     9A 0179     50$:   MOVZBL  #CLMCNX$K_FNC_TRNSTS,R0 ; Facility specific message code
           04 1C A4   08     E1 017C            BBC     #CLUB$V_STS_PPHASE, -     ; Branch if status inquiry phase A
                               0181                    CLUB$L_FLAGS(R4),60$
                 50   18     E2 0181            BBSS    #CLMCNX$V_RP_TRNSTS_CMT+24, - ; Request commitment on PH1B status
                      00        0184                    R0,60$                    ;    -- to go into message reply field
                   118A      30 0185     60$:   BSBW    INIT_STD_MSG              ; Standard CDRP message initialization
                   FE75'     30 0188            BSBW    CNX$SEND_MSG_CSB          ; Send message
                               018B     ;
                               018B     ;       We are resumed here when the response message arrives.
                               018B     ;       Registers contain:
                               018B     ;               R0:     Status
                               018B     ;               R2:     Address of message buffer
                               018B     ;               R3:     Address of CSB
                               018B     ;               R4:     Address of PDT
                               018B     ;               R5:     Address of CDRP
                               018B     ;
                   1158      30 018B            BSBW    CNX$PROCESS_RESPONSE      ; Deallocate storage, get status
                   14 50     E9 018E            BLBC    R0,70$                    ; Branch if NACK or connection broken
                               0191            ASSUME  CLMCNX$K_RP_TRNSTS_PH0+1 EQ CLMCNX$K_RP_TRNSTS_PH1B
                               0191            ASSUME  CLMCNX$K_RP_TRNSTS_PH1B+1 EQ CLMCNX$K_RP_TRNSTS_PH1
```

```
                   0191          ASSUME  CLMCNX$K_RP_TRNSTS_PH1+1 EQ CLMCNX$K_RP_TRNSTS_PH2
                   0191          ASSUME  CLUB$V_STS_PH0+1 EQ CLUB$V_STS_PH1B
                   0191          ASSUME  CLUB$V_STS_PH1B+1 EQ CLUB$V_STS_PH1
                   0191          ASSUME  CLUB$V_STS_PH1+1 EQ CLUB$V_STS_PH2
   50   13 A2   9A 0191          MOVZBL  CLMCNX$B_REPLY(R2),R0
        50 01   C2 0195          SUBL2   #CLMCNX$K_RP_TRNSTS_PH0,R0 ; Base on zero
        04 50   D1 0198          CMPL    R0,#CLMCNX$K_RP_TRNSTS_PH2 ; Check range
           0E   1A 019B          BGTRU   80$                     ; Out of range reply, die
        50 09   C0 019D          ADDL2   #CLUB$V_STS_PH0,R0      ; Compute bit number in CLUB
00 1C A4   50   E2 01A0          BBSS    R0,CLUB$L_FLAGS(R4),70$ ; Set status bit for this node
                   01A5  ;
                   01A5  ; terminate threads -- except for the last one
                   01A5  ;
     5A A4      B7 01A5  70$:    DECW    CLUB$W_MSGCNT(R4)
           05   19 01A8          BLSS    90$                     ; Branch when done
                05 01AA          RSB                             ; Terminate thread
                   01AB
                   01AB  80$:    BUG_CHECK       CNXMGRERR,FATAL ; Inconsistent message data received
                   01AF  ;
                   01AF  ; Get here after all available member nodes have been polled.
                   01AF  ; Evaluate results of the poll and take appropriate action
                   01AF  ;
   50   1C A4   D0 01AF  90$:    MOVL    CLUB$L_FLAGS(R4),R0     ; Get flag bits for easy handling
   08 50   09   E1 01B3          BBC     #CLUB$V_STS_PH0,R0,100$ ; Branch if Phase 0 not seen
   04 50   0C   E1 01B7          BBC     #CLUB$V_STS_PH2,R0,100$ ; Branch if Phase 2 not seen
        FE42'   30 01BB          BSBW    CNX$BUGCHECK_CLUSTER    ; Phase 0 and Phase 2 were seen -- die!
                05 01BE          RSB
                   01BF
   03 50   09   E1 01BF  100$:   BBC     #CLUB$V_STS_PH0,R0,110$ ; Branch if Phase 0 not seen
         1080   31 01C3          BRW     UNLOCK_ALL              ; Some node was in Phase 0 -- Bail out and r
                   01C6
   0B 50   0C   E1 01C6  110$:   BBC     #CLUB$V_STS_PH2,R0,130$ ; Branch if Phase 2 not seen
   03 50   14   E0 01CA          BBS     #CLUB$V_BACKOUT,R0,120$ ; Branch if we have commited to backing out
         0F37   31 01CE          BRW     DO_PHASE2               ; Pretend a phase 2 message has arrived
                   01D1
                   01D1  120$:   BUG_CHECK       CLUEXIT,FATAL   ; Saw phase 2 after commiting to backing out
                   01D5
0A 1C A4   08   E2 01D5  130$:   BBSS    #CLUB$V_STS_PPHASE, -   ; Set polling phase B and do it
                   01DA                  CLUB$L_FLAGS(R4),150$
                   01DA  ;
                   01DA  ; Polling phase A just completed
                   01DA  ;
   03 50   0B   E1 01DA          BBC     #CLUB$V_STS_PH1,R0,140$ ; Phase 1 with good connection not seen
         FF5E   31 01DE          BRW     10$                     ; Repeat first polling phase
         FF63   31 01E1  140$:   BRW     20$                     ; Perform second polling phase
                   01E4
   04 50   0B   E1 01E4  150$:   BBC     #CLUB$V_STS_PH1,R0,160$ ; Phase 1 with good connection not seen
                   01E8          BUG_CHECK       CNXMGRERR,FATAL ; Consistency error -- node with good connec
                   01EC                                          ; seen during polling phase B
                   01EC
         1057   31 01EC  160$:   BRW     UNLOCK_ALL              ; Second phase was unconclusive, back-out
```

```
                                  01EF          .SBTTL  CNX$DISK_CHANGE - Quorum Disk Connection State Change
                                  01EF
                                  01EF  ;++
                                  01EF  ;
                                  01EF  ; FUNCTIONAL DESCRIPTION:
                                  01EF  ;
                                  01EF  ;       This routine is called when the "connection" to the quorum disk has broken
                                  01EF  ;       or has been re-established.
                                  01EF  ;
                                  01EF  ; CALLING SEQUENCE:
                                  01EF  ;
                                  01EF  ;       JSB     CNX$DISK_CHANGE
                                  01EF  ;       Kernel mode, IPL .le. IPL$_SCS = IPL$_SYNCH
                                  01EF  ;
                                  01EF  ; INPUT PARAMETERS:
                                  01EF  ;
                                  01EF  ;       NONE
                                  01EF  ;
                                  01EF  ; OUTPUT PARAMETERS:
                                  01EF  ;
                                  01EF  ;       NONE
                                  01EF  ;
                                  01EF  ; COMPLETION CODES:
                                  01EF  ;
                                  01EF  ;       NONE
                                  01EF  ;
                                  01EF  ; SIDE EFFECTS:
                                  01EF  ;
                                  01EF  ;       R0 and R1 are destroyed
                                  01EF  ;
                                  01EF  ;--
                                  01EF
                                  01EF  CNX$DISK_CHANGE::
                                  01EF          DSBINT  #IPL$_SYNCH                     ; Save and raise IPL
                      3C    BB    01F5          PUSHR   #^M<R2,R3,R4,R5>                ; Save registers
        54   00000000'GF    D0    01F7          MOVL    G^CLU$GL_CLUB,R4                ; Address of CLUB
                53   10 A4  D0    01FE          MOVL    CLUB$L_LOCAL_CSB(R4),R3         ; Address of local CSB
  50    1C A4    01    01   EF    0202          EXTZV   #CLUB$V_QF_ACTIVE,#1, -         ; Extract quorum file active bit
                                  0208                  CLUB$L_FLAGS(R4),R0             ;   from CLUB
  60 A3    01    09    50   F0    0208          INSV    R0, -                           ; Insert quorum file active bit
                                  020E                  #CSB$V_QF_ACTIVE,#1, -          ;   extracted from CLUB into the
                                  020E                  CSB$L_STATUS(R3)                ;   local CSB
                   FDEF'    30    020E          BSBW    CNX$SCAN_CSBS                   ; Iterate over all CSBs
                   06 50    E9    0211          BLBC    R0,30$                          ; Branch when done
          00 60 A3    1A    E2    0214          BBSS    #CSB$V_SEND_STATUS, -           ; Set status request bit
                                  0219                  CSB$L_STATUS(R3),20$
                            05    0219  20$:    RSB
                                  021A
                   FDE3'    30    021A  30$:    BSBW    CNX$CHECK_QUORUM                ; Hang if necessary
                    0335    30    021D          BSBW    SCAN                            ; Look for work to do
                      3C    BA    0220          POPR    #^M<R2,R3,R4,R5>                ; Restore registers
                                  0222          ENBINT                                  ; Restore IPL
                            05    0225          RSB                                     ; Return
```

```
                          0226          .SBTTL  CNX$CHANGE_QUORUM - Adjust quorum
                          0226
                          0226  ;++
                          0226  ;
                          0226  ; FUNCTIONAL DESCRIPTION:
                          0226  ;
                          0226  ;       Request that the cluster quorum be updated to a
                          0226  ;       specified value.  This requires a full cluster state
                          0226  ;       transition to agree on a new value.  The value is
                          0226  ;       adjusted so as not to leave the cluster without a
                          0226  ;       quorum and so as not to risk paritioning given a
                          0226  ;       universe of the nodes currently in the cluster.
                          0226  ;
                          0226  ; CALLING SEQUENCE:
                          0226  ;
                          0226  ;       JSB     CNX$CHANGE_QUORUM
                          0226  ;       IPL is IPL$_SCS
                          0226  ;
                          0226  ;       The requestor can determine that the operator is complete by
                          0226  ;       polling the CLUB$V_ADJ_QUORUM bit in the CLUB$L_FLAGS word.
                          0226  ;       This bit is cleared after the quorum has been adjusted.
                          0226  ;
                          0226  ; INPUT PARAMETERS:
                          0226  ;
                          0226  ;       R1 is new quorum value
                          0226  ;
                          0226  ; OUTPUT PARAMETERS:
                          0226  ;
                          0226  ;       NONE
                          0226  ;
                          0226  ; COMPLETION CODES:
                          0226  ;
                          0226  ;       R0 is success status.
                          0226  ;
                          0226  ; SIDE EFFECTS:
                          0226  ;
                          0226  ;       R1 is destroyed.
                          0226  ;
                          0226  ;--
                          0226
                          0226  CNX$CHANGE_QUORUM::
                 3C   BB  0226          PUSHR   #^M<R2,R3,R4,R5>        ; Save registers
                 50   D4  0228          CLRL    R0                      ; Assume failure
     54   00000000'GF D0  022A          MOVL    G^CLU$GL_CLUB,R4        ; Address of CLUB
       10 1C A4   1D  E0  0231          BBS     #CLUB$V_TRANSITION, -   ; Branch if a state change in progress
                          0236                  CLUB$L_FLAGS(R4),20$
       0B 1C A4   1B  E2  0236          BBSS    #CLUB$V_ADJ_QUORUM, -   ; Set quorum update request bit
                          023B                  CLUB$L_FLAGS(R4),20$    ;   branch if already set
         00A6 C4  51  B0  023B          MOVW    R1, -                   ; Store requested quorum value
                          0240                  CLUB$W_ADJ_QUORUM(R4)
             0312     30  0240  10$:    BSBW    SCAN                    ; Look for work to do
          50   01     3C  0243          MOVZWL  #SS$_NORMAL,R0          ; Return success status
                 3C   BA  0246  20$:    POPR    #^M<R2,R3,R4,R5>        ; Restore registers
                      05  0248          RSB                             ; Return
```

```
                              0249          .SBTTL  CNX$SHUTDOWN - Request cluster shutdown
                              0249
                              0249  ;++
                              0249  ;
                              0249  ; FUNCTIONAL DESCRIPTION:
                              0249  ;
                              0249  ;       This entry point is called to request a cluster shutdown.
                              0249  ;       While in shutdown mode:
                              0249  ;               No nodes may join cluster.
                              0249  ;               If quorum is lost, shutdown immediately.
                              0249  ;               When all systems reach shutdown state, shutdown this node.
                              0249  ;
                              0249  ; CALLING SEQUENCE:
                              0249  ;
                              0249  ;       JSB     CNX$SHUTDOWN
                              0249  ;       IPL is IPL$_SCS
                              0249  ;
                              0249  ; INPUT PARAMETERS:
                              0249  ;
                              0249  ;       NONE
                              0249  ;
                              0249  ; OUTPUT PARAMETERS:
                              0249  ;
                              0249  ;       NONE
                              0249  ;
                              0249  ; COMPLETION CODES:
                              0249  ;
                              0249  ;       R0 is success status
                              0249  ;
                              0249  ; SIDE EFFECTS:
                              0249  ;
                              0249  ;       R1 is destroyed.
                              0249  ;
                              0249  ;--
                              0249
                              0249  CNX$SHUTDOWN::
                3C      BB    0249          PUSHR   #^M<R2,R3,R4,R5>        ; Save registers
  54     00000000'GF    D0    024B          MOVL    G^CLU$GL_CLUB,R4        ; Address of CLUB
           53   10 A4   D0    0252          MOVL    CLUB$L_LOCAL_CSB(R4),R3 ; Local system CSB address
     11 1C A4     02    E2    0256          BBSS    #CLUB$V_SHUTDOWN, -     ; Set shutdown mode in CLUB
                              025B                  CLUB$L_FLAGS(R4),40$    ;  and branch if already set
     00 60 A3     0A    E2    025B          BBSS    #CSB$V_SHUTDOWN, -      ; Set shutdown mode in CSB
                              0260                  CSB$L_STATUS(R3),20$
             FD9D'      30    0260  20$:    BSBW    CNX$SCAN_CSBS           ; Iterate over all CSBs
               06 50    E9    0263          BLBC    R0,40$                  ; Branch when done
     00 60 A3     1A    E2    0266          BBSS    #CSB$V_SEND_STATUS, -   ; Request that status be sent
                              026B                  CSB$L_STATUS(R3),30$
                05            026B  30$:    RSB
                              026C
              02E6      30    026C  40$:    BSBW    SCAN                    ; Look for things to do
          50    01      3C    026F          MOVZWL  #SS$_NORMAL,R0          ; Return success status
                3C      BA    0272          POPR    #^M<R2,R3,R4,R5>        ; Restore registers
                05            0274          RSB                             ; Return
```

```
                        0275        .SBTTL  CNX$SEND_ALL_STATUS - Send Status to All Flagged Systems
                        0275
                        0275  ;++
                        0275  ;
                        0275  ; FUNCTIONAL DESCRIPTION:
                        0275  ;
                        0275  ;       This routine is called to send our status to all systems
                        0275  ;       for whom the send status flag is set.
                        0275  ;       The routine returns synchronously, status will eventually
                        0275  ;       get sent or the link will break and the status will never
                        0275  ;       be sent.
                        0275  ;
                        0275  ; CALLING SEQUENCE:
                        0275  ;
                        0275  ;       JSB     CNX$SEND_ALL_STATUS
                        0275  ;       IPL is IPL$_SCS
                        0275  ;
                        0275  ; INPUT PARAMETERS:
                        0275  ;
                        0275  ;       NONE
                        0275  ;
                        0275  ; OUTPUT PARAMETERS:
                        0275  ;
                        0275  ;       R4 is address of CLUB
                        0275  ;
                        0275  ; COMPLETION CODES:
                        0275  ;
                        0275  ;       R0
                        0275  ;               Success (even): Status will be sent or the link will break
                        0275  ;               Failure (odd): Status cannot be sent, probably because storage
                        0275  ;                       cannot be allocated
                        0275  ;
                        0275  ; SIDE EFFECTS:
                        0275  ;
                        0275  ;       R1 is destroyed
                        0275  ;
                        0275  ;--
                        0275
                        0275  CNX$SEND_ALL_STATUS::
          2C      BB    0275        PUSHR   #^M<R2,R3,R5>            ; Save registers
     55   01      D0    0277        MOVL    #1,R5                    ; Init summary status
        FD83'     30    027A        BSBW    CNX$SCAN_CSBS            ; Look for CSB needing status
       13 50      E9    027D        BLBC    R0, 20$                  ; Branch when done
 0D 60 A3 00      E0    0280        BBS     #CSB$V_LONG_BREAK, -     ; Ignore nodes with permanently
                        0285                CSB$L_STATUS(R3),10$     ;   connections
 08 60 A3 1A      E1    0285        BBC     #CSB$V_SEND_STATUS, -    ; Branch if status not requested
                        028A                CSB$L_STATUS(R3),10$
          0D      10    028A        BSBB    SEND_STATUS              ; Queue status message
       03 50      E8    028C        BLBS    R0,10$                   ; Branch on success
     55   50      D0    028F        MOVL    R0,R5                    ; Save status
                  05    0292  10$:  RSB
                        0293
     50   55      D0    0293  20$:  MOVL    R5,R0                    ; Fetch status
          2C      BA    0296        POPR    #^M<R2,R3,R5>            ; Restore registers
                  05    0298        RSB                              ; Return
```

```
                        0299         .SBTTL  SEND_STATUS - Send Status to a System
                        0299
                        0299   ;++
                        0299   ;
                        0299   ; FUNCTIONAL DESCRIPTION:
                        0299   ;
                        0299   ;       This routine is called to send our status to a system.
                        0299   ;       The routine returns synchronously, status will eventually
                        0299   ;       get sent or the link will break and the status will never
                        0299   ;       be sent.
                        0299   ;
                        0299   ; CALLING SEQUENCE:
                        0299   ;
                        0299   ;       JSB     SEND_STATUS
                        0299   ;       IPL is IPL$_SCS
                        0299   ;
                        0299   ; INPUT PARAMETERS:
                        0299   ;
                        0299   ;       R3: CSB of new system
                        0299   ;
                        0299   ; OUTPUT PARAMETERS:
                        0299   ;
                        0299   ;       NONE
                        0299   ;
                        0299   ; COMPLETION CODES:
                        0299   ;
                        0299   ;       R0
                        0299   ;               Success (even): Status will be sent or the link will break
                        0299   ;               Failure (odd): Status cannot be sent, probably because storage
                        0299   ;                       cannot be allocated
                        0299   ;
                        0299   ; SIDE EFFECTS:
                        0299   ;
                        0299   ;       R1 is destroyed
                        0299   ;
                        0299   ;--
                        0299
                        0299   SEND_STATUS:
            3C     BB   0299           PUSHR   #^M<R2,R3,R4,R5>           ; Save registers
   23 60 A3   18   E0   029B           BBS     #CSB$V_LOCAL, -            ; Skip if local system
                        02A0                   CSB$L_STATUS(R3),20$
   1E 60 A3   00   E0   02A0           BBS     #CSB$V_LONG_BREAK, -       ; Skip if long break
                        02A5                   CSB$L_STATUS(R3),20$
          FD58'    30   02A5           BSBW    CNX$ALLOC_CDRP_ONLY
          1B 50    E9   02A8           BLBC    R0,30$                     ; No memory available
   4C A5   C9'AF   9E   02AB           MOVAB   B^BUILD_STS_MSG, -         ; Address of routine to build
                        02B0                   CDRP$L_MSGBLD(R5)          ;  status message
   00 60 A3   1A   E5   02B0           BBCC    #CSB$V_SEND_STATUS, -      ; Clear "send status" request
                        02B5                   CSB$L_STATUS(R3),10$
          FD48'    30   02B5   10$:    BSBW    CNX$SEND_FORGET            ; Setup immediate return
         55  53    D0   02B8           MOVL    R3,R5                      ; Address of CSB
   50  0000'CF     9E   02BB           MOVAB   SNDSTS_MSG,R0              ; Sending status message text
          FD3D'    30   02C0           BSBW    CNX$CONFIG_CHANGE          ; Make this state known
      50   01      D0   02C3   20$:    MOVL    S^#SS$_NORMAL,R0           ; Return success
            3C     BA   02C6   30$:    POPR    #^M<R2,R3,R4,R5>           ; Restore registers
            05          02C8           RSB                                ; Return
                        02C9
```

```
                        02C9             .SBTTL  BLD_STS_MSG - Build Status Message
                        02C9
                        02C9      ;++
                        02C9      ;
                        02C9      ; FUNCTIONAL DESCRIPTION:
                        02C9      ;
                        02C9      ;       This routine builds a status message describing the local node and the
                        02C9      ;       cluster of which it is a member.
                        02C9      ;
                        02C9      ; CALLING SEQUENCE:
                        02C9      ;
                        02C9      ;       JSB     BLD_STS_MSG
                        02C9      ;       IPL is IPL$_SCS
                        02C9      ;
                        02C9      ; INPUT PARAMETERS:
                        02C9      ;
                        02C9      ;       R2:     Address of message buffer
                        02C9      ;       R3:     Address of CSB
                        02C9      ;       R4:     Address of PDT
                        02C9      ;       R5:     Address of CDRP
                        02C9      ;
                        02C9      ; OUTPUT PARAMETERS:
                        02C9      ;
                        02C9      ;       NONE
                        02C9      ;
                        02C9      ; COMPLETION CODES:
                        02C9      ;
                        02C9      ;       NONE
                        02C9      ;
                        02C9      ; SIDE EFFECTS:
                        02C9      ;
                        02C9      ;       R0 and R1 are destroyed
                        02C9      ;
                        02C9      ;--
                        02C9
                        02C9      BUILD_STS_MSG:
   08 A2     01    90   02C9              MOVB    #CLSMSG$K_FAC_CNX, -      ; Facility identifier
                        02CD                      CLSMSG$B_FACILITY(R2)
   09 A2     01    90   02CD              MOVB    #CLMCNX$K_FNC_STATUS, - ; Facility specific function code
                        02D1                      CLSMSG$B_FUNC(R2)
   51     64 A3    D0   02D1              MOVL    CSB$L_CLUB(R3),R1         ; Address of CLUB
          0C A2    94   02D5              CLRB    CLMSTS$B_FLAGS(R2)        ; Clear flags bits
04 1C A1     00    E1   02D8              BBC     #CLUB$V_CLUSTER, -        ; Branch if bit not set
                        02DD                      CLUB$L_FLAGS(R1),10$
   0C A2     01    88   02DD              BISB    #CLMSTS$M_CLUSTER, -      ; Set bit in message
                        02E1                      CLMSTS$B_FLAGS(R2)
04 1C A1     01    E1   02E1      10$:    BBC     #CLUB$V_QF_ACTIVE, -      ; Branch if bit not set
                        02E6                      CLUB$L_FLAGS(R1),20$
   0C A2     02    88   02E6              BISB    #CLMSTS$M_QF_ACTIVE, -    ; Set bit in message
                        02EA                      CLMSTS$B_FLAGS(R2)
04 1C A1     02    E1   02EA      20$:    BBC     #CLUB$V_SHUTDOWN, -       ; Branch if bit not set
                        02EF                      CLUB$L_FLAGS(R1),30$
   0C A2     04    88   02EF              BISB    #CLMSTS$M_SHUTDOWN, -     ; Set bit in message
                        02F3                      CLMSTS$B_FLAGS(R2)
0E A2     20 A1    B0   02F3      30$:    MOVW    CLUB$W_QUORUM(R1), -      ; Cluster quorum
                        02F8                      CLMSTS$W_CQUORUM(R2)
10 A2     22 A1    B0   02F8              MOVW    CLUB$W_VOTES(R1), -       ; Cluster votes
```

```
                          02FD              CLMSTS$W_CVOTES(R2)
     12 A2   24 A1   B0   02FD      MOVW    CLUB$W_NODES(R1), -       ; Cluster nodes
                          0302              CLMSTS$W_NODES(R2)
        50   10 A1   D0   0302      MOVL    CLUB$L_LOCAL_CSB(R1),R0 ; Address of local CSB
     14 A2   52 A0   B0   0306      MOVW    CSB$W_QUORUM(R0), -       ; Node quorum
                          030B              CLMSTS$W_NQUORUM(R2)
     16 A2   50 A0   B0   030B      MOVW    CSB$W_VOTES(R0), -        ; Nodes votes
                          0310              CLMSTS$W_NVOTES(R2)
     1C A2   2C A1   7D   0310      MOVQ    CLUB$Q_FTIME(R1), -       ; Cluster formation time
                          0315              CLMSTS$Q_FTIME(R2)
     24 A2   3C A1   7D   0315      MOVQ    CLUB$Q_LST_TIME(R1), -    ; Time stamp of last transaction
                          031A              CLMSTS$Q_LST_TIME(R2)
     18 A2   56 A0   B0   031A      MOVW    CSB$W_QDVOTES(R0), -      ; Quorum disk votes proposed by local
                          031F              CLMSTS$W_QDVOTES(R2)      ;   node
     1A A2   54 A0   B0   031F      MOVW    CSB$W_LCKDIRWT(R0), -     ; Lock manager directory system weight
                          0324              CLMSTS$W_LCKDIRWT(R2)
     2C A2   68 A1   D0   0324      MOVL    CLUB$L_MAX_XTN(R1), -     ; Largest transaction code
                          0329              CLMSTS$L_MAX_XTN(R2)      ;  seen
     40 A2   74 A0   7D   0329      MOVQ    CSB$Q_REFTIME(R0), -      ; Put local reference time into message
                          032E              CLMSTS$Q_REFTIME(R2)
                  3C BB   032E      PUSHR   #^M<R2,R3,R4,R5>          ; Save registers
30 A2   00B8 C1   10 28   0330      MOVC3   #CLMSTS$S_QDISK, -        ; Quorum disk name
                          0337              CLUB$T_QDNAME(R1), -      ; from SYSGEN parameter
                          0337              CLMSTS$B_QDISK(R2)
                  3C BA   0337      POPR    #^M<R2,R3,R4,R5>          ; Restore registers
                     05   0339      RSB
```

```
                                 033A         .SBTTL  CNX$RCVD_STATUS - Status Received from a System
                                 033A
                                 033A   ;++
                                 033A   ;
                                 033A   ; FUNCTIONAL DESCRIPTION:
                                 033A   ;
                                 033A   ;       This routine is called when a status message is received.
                                 033A   ;       Data from the message is copied in the CSB.  A cluster
                                 033A   ;       transition may be initiated as a result of seeing the
                                 033A   ;       message.
                                 033A   ;
                                 033A   ; CALLING SEQUENCE:
                                 033A   ;
                                 033A   ;       JSB     CNX$RCVD_STATUS
                                 033A   ;       IPL is IPL$_SCS
                                 033A   ;
                                 033A   ; INPUT PARAMETERS:
                                 033A   ;
                                 033A   ;       R2:     Message address
                                 033A   ;       R3:     CSB of sending system
                                 033A   ;       R4:     PDT address
                                 033A   ;
                                 033A   ; OUTPUT PARAMETERS:
                                 033A   ;
                                 033A   ;       NONE
                                 033A   ;
                                 033A   ; COMPLETION CODES:
                                 033A   ;
                                 033A   ;       NONE
                                 033A   ;
                                 033A   ; SIDE EFFECTS:
                                 033A   ;
                                 033A   ;       R0-R5 may be destroyed.
                                 033A   ;--
                                 033A
                                 033A   CNX$RCVD_STATUS::
              55    53    D0     033A           MOVL    R3,R5                   ; Address of CSB
        50    0000'CF     9E     033D           MOVAB   RCVSTS_MSG,R0           ; Sending status message text
               FCBB'      30     0342           BSBW    CNX$CONFIG_CHANGE       ; Make this state known
                   0C     BB     0345           PUSHR   #^M<R2,R3>              ; Save needed registers
         54    64 A3      D0     0347           MOVL    CSB$L_CLUB(R3),R4       ; Address of CLUB
60 A3   00000700 8F       CA     034B           BICL2   #<CSB$M_CLUSTER ! -     ; Clear out received status bits in CSB
                                 0353                     CSB$M_QF_ACTIVE ! -
                                 0353                     CSB$M_SHUTDOWN>, -
                                 0353                   CSB$L_STATUS(R3)
      05 0C A2    00      E1     0353           BBC     #CLMSTS$V_CLUSTER, -    ; Branch if bit clear
                                 0358                   CLMSTS$B_FLAGS(R2),1$
      00 60 A3    08      E2     0358           BBSS    #CSB$V_CLUSTER, -       ; Set bit in CSB
                                 035D                   CSB$L_STATUS(R3),1$
      05 0C A2    01      E1     035D   1$:     BBC     #CLMSTS$V_QF_ACTIVE, -  ; Branch if bit clear
                                 0362                   CLMSTS$B_FLAGS(R2),2$
      00 60 A3    09      E2     0362           BBSS    #CSB$V_QF_ACTIVE, -     ; Set bit in CSB
                                 0367                   CSB$L_STATUS(R3),2$
      05 0C A2    02      E1     0367   2$:     BBC     #CLMSTS$V_SHUTDOWN, -   ; Branch if bit clear
                                 036C                   CLMSTS$B_FLAGS(R2),3$
      00 60 A3    0A      E2     036C           BBSS    #CSB$V_SHUTDOWN, -      ; Set bit in CSB
                                 0371                   CSB$L_STATUS(R3),3$
```

```
      00 60 A3    19  E2  0371  3$:     BBSS    #CSB$V_STATUS_RCVD, -    ; Add status received flag
                          0376                  CSB$L_STATUS(R3),4$
      19 60 A3    01  E0  0376  4$:     BBS     #CSB$V_MEMBER, -         ; Branch if remote node is a member
                          037B                  CSB$L_STATUS(R3),5$      ;   of the local cluster
      52 A3    14 A2  B0  037B          MOVW    CLMSTS$W_NQUORUM(R2), -  ; Remote node's quorum
                          0380                  CSB$W_QUORUM(R3)
      50 A3    16 A2  B0  0380          MOVW    CLMSTS$W_NVOTES(R2), -   ; Remote node's votes
                          0385                  CSB$W_VOTES(R3)
      6E A3    12 A2  B0  0385          MOVW    CLMSTS$W_NODES(R2), -    ; Number of nodes in remote
                          038A                  CSB$W_NODES(R3)          ;   node's cluster
      56 A3    18 A2  B0  038A          MOVW    CLMSTS$W_QDVOTES(R2), -  ; Proposed votes for quorum disk
                          038F                  CSB$W_QDVOTES(R3)
      54 A3    1A A2  B0  038F          MOVW    CLMSTS$W_LCKDIRWT(R2), - ; Lock manager directory system weight
                          0394                  CSB$W_LCKDIRWT(R3)
   50 68 A4    2C A2  C3  0394  5$:     SUBL3   CLMSTS$L_MAX_XTN(R2), -  ; Update largest transaction seen
                          039A                  CLUB$L_MAX_XTN(R4),R0
                  05  18  039A          BGEQ    10$                      ; No update needed
      68 A4    2C A2  D0  039C          MOVL    CLMSTS$L_MAX_XTN(R2), -  ; Update local information
                          03A1                  CLUB$L_MAX_XTN(R4)
      50   00B8 C4    9A  03A1  10$:    MOVZBL  CLUB$T_QDNAME(R4),R0     ; Is local name defined?
                  1D  13  03A6          BEQL    20$                      ; If equal, then can't be the same
         30 A2    50  91  03A8          CMPB    R0,CLMSTS$B_QDISK(R2)    ; Compare to received name length
                  17  12  03AC          BNEQ    20$                      ; Branch if different
                  0C  BB  03AE          PUSHR   #^M<R2,R3>               ; Save registers
 31 A2   00B9 C4  50  29  03B0          CMPC3   R0,CLUB$T_QDNAME+1(R4), - ; Quorum disk names match?
                          03B7                  CLMSTS$B_QDISK+1(R2)
                  0C  BA  03B7          POPR    #^M<R2,R3>               ; Restore registers
                  0A  12  03B9          BNEQ    20$                      ; Branch if disk names differ
      05 60 A3    03  E2  03BB          BBSS    #CSB$V_QF_SAME, -        ; Remember that disk names match
                          03C0                  CSB$L_STATUS(R3),20$     ;   and branch if already known to match
      00 60 A3    1A  E2  03C0          BBSS    #CSB$V_SEND_STATUS, -    ; Just set same bit, send status so
                          03C5                  CSB$L_STATUS(R3),20$     ;   remote node has same opportunity
                          03C5  20$:
      05 60 A3    01  E0  03C5          BBS     #CSB$V_MEMBER, -         ; Branch if remote node is a
                          03CA                  CSB$L_STATUS(R3),30$     ;   member of the local cluster
      74 A3    40 A2  7D  03CA          MOVQ    CLMSTS$Q_REFTIME(R2), -  ; Store reference time
                          03CF                  CSB$Q_REFTIME(R3)
                          03CF  30$:
                          03CF  ;
                          03CF  ; Dispatch on state of this node
                          03CF  ;
      6E 1C A4    13  E1  03CF          BBC     #CLUB$V_INIT, -          ; Branch to exit if initialization
                          03D4                  CLUB$L_FLAGS(R4),300$    ;   flag is not yet set
      05 60 A3    01  E1  03D4          BBC     #CSB$V_MEMBER, -         ; Branch if remote node is not
                          03D9                  CSB$L_STATUS(R3),100$    ; a member of the local cluster
                          03D9  ;
                          03D9  ; The local node is part of an existing cluster.
                          03D9  ; Update quorum calculation to reflect any change seen by the remote node,
                          03D9  ; for example a change in status of the quorum disk.
                          03D9  ;
               FC24'  30  03D9          BSBW    CNX$CHECK_QUORUM         ; Perform dynamic quorum check
                  64  11  03DC          BRB     300$                     ; Branch to common exit
                          03DE
      4D 1C A4    00  E1  03DE  100$:   BBC     #CLUB$V_CLUSTER, -       ; Branch if local node is not
                          03E3                  CLUB$L_FLAGS(R4), 200$   ; a cluster member
                          03E3  ;
                          03E3  ; Local node is a cluster member.
```

```
                          03E3      ;
                          03E3      ; If remote node is a member of another cluster, bugcheck one of the
                          03E3      ; clusters.  It is illegal for two clusters ever to see each other.
                          03E3      ;
     5A 60 A3    08   E1  03E3              BBC     #CSB$V_CLUSTER, -          ; Branch if remote node is not a
                          03E8                      CSB$L_STATUS(R3), 300$     ; member of another cluster
     20 A2    30 A4   D1  03E8              CMPL    CLUB$Q_FTIME+4(R4), -      ; Compare high order foundation times
                          03ED                      CLMSTS$Q_FTIME+4(R2)
              35      1A  03ED              BGTRU   110$                       ; We are younger, so we die
              38      1F  03EF              BLSSU   120$                       ; Other is younger and should die
     1C A2    2C A4   D1  03F1              CMPL    CLUB$Q_FTIME(R4), -        ; Compare low order foundation times
                          03F6                      CLMSTS$Q_FTIME(R2)
              2C      1A  03F6              BGTRU   110$                       ; We are younger, so we die
              2F      1F  03F8              BLSSU   120$                       ; Other is younger and should die
     28 A2    40 A4   D1  03FA              CMPL    CLUB$Q_LST_TIME+4(R4), -   ; Compare high order transition times
                          03FF                      CLMSTS$Q_LST_TIME+4(R2)
              23      1A  03FF              BGTRU   110$                       ; We are younger, so we die
              26      1F  0401              BLSSU   120$                       ; Other is younger and should die
     24 A2    3C A4   D1  0403              CMPL    CLUB$Q_LST_TIME(R4), -     ; Compare low order transition times
                          0408                      CLMSTS$Q_LST_TIME(R2)
              1A      1A  0408              BGTRU   110$                       ; We are younger, so we die
              1D      1F  040A              BLSSU   120$                       ; Other is younger and should die
              0C      BB  040C              PUSHR   #^M<R2,R3>                 ; Save registers
        52    10 A4   D0  040E              MOVL    CLUB$L_LOCAL_CSB(R4),R2    ; Local CSB address
        52    68 A2   D0  0412              MOVL    CSB$L_SB(R2),R2            ; SB for local system
        53    68 A3   D0  0416              MOVL    CSB$L_SB(R3),R3            ; SB for remote system
18 A3    18 A2    06  29  041A              CMPC3   #SB$S_SYSTEMID, -          ; Compare system ID's
                          0420                      SB$B_SYSTEMID(R2), -
                          0420                      SB$B_SYSTEMID(R3)
              0C      BA  0420              POPR    #^M<R2,R3>                 ; Restore registers
              05      1F  0422              BLSSU   120$                       ; Other should die
                          0424      110$:
                          0424      ;
                          0424      ; Should put code in R0 to explain the bugcheck
                          0424      ;
           FBD9'      30  0424              BSBW    CNX$BUGCHECK_CLUSTER       ; Bring down local cluster
              19      11  0427              BRB     300$                       ; Exit
                          0429      ;
                          0429      ; Get here if we believe that the other cluster should die.
                          0429      ; Send it status which may help it to realize this.
                          0429      ;
     00 60 A3    1A   E2  0429      120$:   BBSS    #CSB$V_SEND_STATUS, -      ; Request that status be sent
                          042E                      CSB$L_STATUS(R3),130$
              12      11  042E      130$:   BRB     300$                       ; Branch to common exit
                          0430      ;
                          0430      ; Get here if local node is not part of a cluster
                          0430      ;
        01    58 A4   91  0430      200$:   CMPB    CLUB$B_CUR_CODE(R4), -     ; Is this node forming a cluster?
                          0434                      #CLMCNX$K_XTN_FORM
              0A      12  0434              BNEQ    210$                       ; Branch if not
     05 0C A2    00   E1  0436              BBC     #CLMSTS$V_CLUSTER, -       ; Branch if remote node is not
                          043B                      CLMSTS$B_FLAGS(R2),210$    ;  a cluster member
     00 1C A4    11   E2  043B              BBSS    #CLUB$V_UNLOCK, -          ; Request failure/unlock to
                          0440                      CLUB$L_FLAGS(R4),210$      ;  abort cluster formation
              00      11  0440      210$:   BRB     300$                       ; Branch to common exit
                          0442
                          0442      ;
```

```
                 0442  1253 ; Common exit after receiving a status message.
                 0442  1254 ; Deallocate message buffer, prepare to scan for work to do, and return
                 0442  1255 ;
      0C    BA   0442  1256 300$:   POPR    #^M<R2,R3>              ; Restore registers
   FBB9'    30   0444  1257         BSBW    CNX$DEALL_MSG_BUF_CSB   ; Deallocate message buffer
      6D    10   0447  1258         BSBB    START_TIMEOUT           ; Look for something to do
            05   0449  1259         RSB
```

```
                              044A          .SBTTL  CNX$RCVD_TRNSTS - Received Transition Status Request Message
                              044A
                              044A     ;++
                              044A     ;
                              044A     ; FUNCTIONAL DESCRIPTION:
                              044A     ;
                              044A     ;       This routine is called when a transition status request message
                              044A     ;       is received.  A reply is sent indicating whether or not the specified
                              044A     ;       transition in in or prior to Phase 0, in or post to Phase 2, or in Phase
                              044A     ;       1 while this node does/doesn't have a connection to the coordinator.
                              044A     ;       The purpose of this message is to allow a node caught in Phase 1 when
                              044A     ;       its connection to the coordinator breaks to figure out what to do.
                              044A     ;
                              044A     ; CALLING SEQUENCE:
                              044A     ;
                              044A     ;       JSB     CNX$RCVD_TRNSTS
                              044A     ;       IPL is IPL$_SCS
                              044A     ;
                              044A     ; INPUT PARAMETERS:
                              044A     ;
                              044A     ;       R2:     Message address
                              044A     ;       R3:     CSB of sending system
                              044A     ;       R4:     PDT address
                              044A     ;       R5:     CDRP address (uninitialized)
                              044A     ;
                              044A     ; OUTPUT PARAMETERS:
                              044A     ;
                              044A     ;       NONE
                              044A     ;
                              044A     ; COMPLETION CODES:
                              044A     ;
                              044A     ;       NONE
                              044A     ;
                              044A     ; SIDE EFFECTS:
                              044A     ;
                              044A     ;       R0-R5 may be destroyed.
                              044A     ;--
                              044A
                              044A     CNX$RCVD_TRNSTS::
                   52    DD   044A          PUSHL   R2                          ; Save address of message
                 FBB1'   30   044C          BSBW    CNX$INIT_CDRP               ; Initialize the CDRP for the response
                   04    BA   044F          POPR    #^M<R2>                     ; Restore address of message
          54    64 A3    D0   0451          MOVL    CSB$L_CLUB(R3),R4           ; Address of CLUB
  50    34 A4    0C A2   C3   0455          SUBL3   CLMCNX$L_XTN_ID(R2), -      ; Compare transition ID in message with
                              045B                  CLUB$L_LST_XTN(R4),R0       ;   last completed transition
                   4E    14   045B          BGTR    90$                         ; Branch if inconsistent
                   2B    13   045D          BEQL    20$                         ; Branch if same and return Phase 2
                              045F     ;
                              045F     ; Here if ID in message is greater than last completed ID
                              045F     ;
  50    48 A4    0C A2   C3   045F          SUBL3   CLMCNX$L_XTN_ID(R2), -      ; Compare transition ID in message with
                              0465                  CLUB$L_CUR_XTN(R4),R0       ;   current transition
                   44    19   0465          BLSS    90$                         ; Branch if inconsistent
                   1D    14   0467          BGTR    10$                         ; Branch if same and return Phase 0
                              0469     ;
                              0469     ; Here if ID in message matches current/last unsuccessful ID
                              0469     ;
```

```
50 8F    59 A4   91  0469         CMPB    CLUB$B_CUR_PHASE(R4), - ; Is phase UNLOCK?
                     046E                 #CLMCNX$K_UNLOCK
            16   13  046E         BEQL    10$                     ; Branch if UNLOCK and return Phase 0
40 8F    59 A4   91  0470         CMPB    CLUB$B_CUR_PHASE(R4), - ; Is phase before or after Phase 1?
                     0475                 #CLMCNX$K_PH1
            0F   19  0475         BLSS    10$                     ; Branch if before phase 1 and return Phase
            11   14  0477         BGTR    20$                     ; Branch if after phase 1 and return Phase 2
                     0479 ;
                     0479 ; Here if local system is in phase 1 of a transition
                     0479 ;
   51    5C A4   D0  0479         MOVL    CLUB$L_COORD(R4),R1     ; CSB of coordinator
0C 60 A1    00   E0  047D         BBS     #CSB$V_LONG_BREAK, -    ; Branch if broken and return Phase 1B
                     0482                 CSB$L_STATUS(R1),30$
            03   DD  0482         PUSHL   #CLMCNX$K_RP_TRNSTS_PH1 ; Return Phase 1
            14   11  0484         BRB     40$                     ; Branch to common exit
                     0486
            01   DD  0486 10$:    PUSHL   #CLMCNX$K_RP_TRNSTS_PH0 ; Return Phase 0
            10   11  0488         BRB     40$                     ; Branch to common exit
                     048A
            04   DD  048A 20$:    PUSHL   #CLMCNX$K_RP_TRNSTS_PH2 ; Return Phase 2
            0C   11  048C         BRB     40$                     ; Branch to common exit
                     048E
            02   DD  048E 30$:    PUSHL   #CLMCNX$K_RP_TRNSTS_PH1B ; Return Phase 1 (coodinator connection bro
05 13 A2    00   E1  0490         BBC     #CLMCNX$V_RP_TRNSTS_CMT, - ; Branch if commitment not requested
                     0495                 CLMCNX$B_REPLY(R2),40$
00 1C A4    14   E2  0495         BBSS    #CLUB$V_BACKOUT, -      ; Set backout inhibit flag in CLUB
                     049A                 CLUB$L_FLAGS(R4),40$
      50    05   9A  049A 40$:    MOVZBL  #CLMCNX$K_FNC_DESC,R0   ; Facility specific function code
          0E69   30  049D         BSBW    CNX$INIT_STD_RESP       ; Init CDRP for standard response
   36 A5    01   90  04A0         MOVB    #1,CDRP$L_VAL3+2(R5)    ; Store success in ACK field
   37 A5    8E   F6  04A4         CVTLB   (SP)+,CDRP$L_VAL3+3(R5) ; Store reply data
         FB55'   31  04A8         BRW     CNX$RESP_FORGET         ; Send response message and forget it
                     04AB
                     04AB 90$:    BUG_CHECK       CNXMGRERR,FATAL ; Inconsistent state of message and this sys
```

```
                          04AF          .SBTTL  START_TIMEOUT - Start random CLUB-based timeout
                          04AF          .SBTTL  START_LONG_TIMEOUT - Start long random CLUB-based timeout
                          04AF
                          04AF  ;++
                          04AF  ;
                          04AF  ; FUNCTIONAL DESCRIPTION:
                          04AF  ;
                          04AF  ;       If the CLUB fork block is free, start a timeout.
                          04AF  ;       An immediate return is made to the caller.  When the
                          04AF  ;       timout occurs, the thread jumps to SCAN.
                          04AF  ;
                          04AF  ; CALLING SEQUENCE:
                          04AF  ;
                          04AF  ;       JSB     START_TIMEOUT
                          04AF  ;       JSB     START_LONG_TIMEOUT
                          04AF  ;       IPL is IPL$_SCS
                          04AF  ;
                          04AF  ; INPUT PARAMETERS:
                          04AF  ;
                          04AF  ;       NONE
                          04AF  ;
                          04AF  ; OUTPUT PARAMETERS:
                          04AF  ;
                          04AF  ;       NONE
                          04AF  ;
                          04AF  ; COMPLETION CODES:
                          04AF  ;
                          04AF  ;       NONE
                          04AF  ;
                          04AF  ; SIDE EFFECTS:
                          04AF  ;
                          04AF  ;       R0-R5 are destroyed
                          04AF  ;
                          04AF  ;--
                          04AF
                          04AF  START_LONG_TIMEOUT:
 50    4E20 8F      3C    04AF          MOVZWL  #20000,R0                       ; 2000...19999 ms timeout
            05      11    04B4          BRB     DO_TIMEOUT
                          04B6
                          04B6  START_TIMEOUT:
 50    1388 8F      3C    04B6          MOVZWL  #5000,R0                        ; Range is 500..4999 milli-seconds
                          04BB
                          04BB  DO_TIMEOUT:
            50      DD    04BB          PUSHL   R0                              ; Save timeout range
          FDB5      30    04BD          BSBW    CNX$SEND_ALL_STATUS             ; Send status as necessary
 53    6E   0A      C7    04C0          DIVL3   #10,(SP),R3                     ; Make range max/10...max-1
 50    8E   53      C3    04C4          SUBL3   R3,(SP)+,R0                     ; Update random number range
          FB35'     30    04C8          BSBW    CNX$RANDOM                      ; Generate a random number
       53   50      C0    04CB          ADDL2   R0,R3                           ; Save timeout in R3
15 1C A4    1D      E0    04CE          BBS     #CLUB$V_TRANSITION, -           ; Branch if activity in progress
                          04D3                  CLUB$L_FLAGS(R4),30$
 55    00CC C4      9E    04D3          MOVAB   CLUB$B_FORK_BLOCK(R4),R5 ; Address of transition fork block
0B 1C A5    00      E0    04D8          BBS     #CLUBFKB$V_FKB_BUSY, -          ; Branch if fork block in use
                          04DD                  CLUBFKB$L_STATUS(R5),30$
       0084 C4      D5    04DD          TSTL    CLUB$L_TQE(R4)                  ; TQE in use?
            05      12    04E1          BNEQ    30$
01 1C A4    13      E0    04E3          BBS     #CLUB$V_INIT, -                 ; Branch if initialization complete
```

```
                            04E8        CLUB$L_FLAGS(R4),40$
                        05  04E8  30$:  RSB
                            04E9
                            04E9  40$:
                            04E9  ;
                            04E9  ; Do timeout, time in milli-seconds in R3
                            04E9  ;
             51    30   3C  04E9  50$:  MOVZWL  #TQE$K_LENGTH,R1          ; Size of timer queue entry
          00000000'GF   16  04EC        JSB     G^EXE$ALONONPAGED         ; Allocate one
                FB0B'   30  04F2        BSBW    CNX$RESOURCE_CHECK        ; Watch out for totally exhausted resources
                3A 50   E9  04F5        BLBC    R0,70$                    ; Can't allocate memory
          08 A2    51   B0  04F8        MOVW    R1,TQE$W_SIZE(R2)         ; Store size
          0A A2    0F   90  04FC        MOVB    #DYN$C_TQE,TQE$B_TYPE(R2) ; Store type
          14 A2    54   D0  0500        MOVL    R4,TQE$L_FR4(R2)          ; Store time delay and CLUB address
        0084 C4    52   D0  0504        MOVL    R2,CLUB$L_TQE(R4)         ; Save TQE address in CLUB
             55    52   D0  0509        MOVL    R2,R5                     ; Move address of TQE
          0B A5    01   90  050C        MOVB    #TQE$C_SSSNGL,TQE$B_RQTYPE(R5) ; Store type of timer queue entry
       0C A5   37'AF    9E  0510        MOVAB   B^80$,TQE$L_FPC(R5)       ; Store address of timer fork process
52  00 00002710 8F  53  7A  0515        EMUL    R3,#10*1000,#0,R2         ; Get milli-seconds and cvt to 100 ns. units
     50   00000000'GF   7D  051E        MOVQ    G^EXE$GQ_SYSTIME,R0       ; Get current time
             50    52   C0  0525        ADDL    R2,R0                     ; Add to current time
             51    53   D8  0528        ADWC    R3,R1
          00000000'GF   16  052B        JSB     G^EXE$INSTIMQ             ; Insert in timer queue
                        05  0531        RSB                               ; Return
                            0532
                FACB'   30  0532  70$:  BSBW    CNX$CLUB_WAIT             ; Wait a second or so in the CLUB
                   B2   11  0535        BRB     50$
                            0537
                            0537  ; Come here as a timer fork process to proceed
                            0537  ; Inputs:
                            0537  ;     R4      CLUB address
                            0537  ;     R5      TQE address
                            0537
             50    55   D0  0537  80$:  MOVL    R5,R0                     ; Address of timer queue entry
          00000000'GF   16  053A        JSB     G^EXE$DEANONPAGED         ; Deallocate it
        0084 C4    55   D1  0540        CMPL    R5,CLUB$L_TQE(R4)         ; Is this timeout wanted?
                   06   12  0545        BNEQ    90$                       ; Branch if not and drop it
              0084 C4   D4  0547        CLRL    CLUB$L_TQE(R4)            ; Clear pointer to TQE
                   08   10  054B        BSBB    SCAN                      ; Set up return address, look for work
     55   00000000'GF   DE  054D  90$:  MOVAL   G^EXE$AL_TQENOREPT,R5     ; Use non-repeating timer queue entry
                        05  0554        RSB
```

```
                        0555        .SBTTL  SCAN - Look for work to do
                        0555
                        0555  ;++
                        0555  ;
                        0555  ; FUNCTIONAL DESCRIPTION:
                        0555  ;
                        0555  ;       This routine is called whenever it might be appropriate to
                        0555  ;       look for something to do.  These instances include timeouts
                        0555  ;       during initialization and after receiving status from nodes.
                        0555  ;
                        0555  ; CALLING SEQUENCE:
                        0555  ;
                        0555  ;       JSB     SCAN
                        0555  ;       IPL is IPL$_SCS
                        0555  ;
                        0555  ; INPUT PARAMETERS:
                        0555  ;
                        0555  ;       NONE
                        0555  ;
                        0555  ; OUTPUT PARAMETERS:
                        0555  ;
                        0555  ;       NONE
                        0555  ;
                        0555  ; COMPLETION CODES:
                        0555  ;
                        0555  ;       NONE
                        0555  ;
                        0555  ; SIDE EFFECTS:
                        0555  ;
                        0555  ;       R0-R5 may be destroyed.
                        0555  ;--
                        0555
          FD1D    30    0555  SCAN:   BSBW    CNX$SEND_ALL_STATUS     ; Send status as needed
         70 50    E9    0558          BLBC    R0,160$                 ; Exit if couldn't send status
 6E 1C A4   13    E1    055B          BBC     #CLUB$V_INIT, -         ; Branch if SYSINIT initialization incomplet
                        0560                  CLUB$L_FLAGS(R4),170$
 69 1C A4   1D    E0    0560          BBS     #CLUB$V_TRANSITION, -   ; Branch if activity in progress
                        0565                  CLUB$L_FLAGS(R4),170$   ;   and drop thread
  55    00CC C4   9E    0565          MOVAB   CLUB$B_FORK_BLOCK(R4),R5 ; Address of transition fork block
 5F 1C A5   00    E0    056A          BBS     #CLUBFKB$V_FKB_BUSY, -  ; Branch if fork block in use
                        056F                  CLUBFKB$L_STATUS(R5),170$
 5B 1C A4   00    E0    056F          BBS     #CLUB$V_CLUSTER, -      ; Branch if this node is cluster member
                        0574                  CLUB$L_FLAGS(R4),200$
                        0574  ;
                        0574  ; Local node is not a member of a cluster.
                        0574  ; Investigate joining or forming a cluster.
                        0574  ;
          FA89'   30    0574          BSBW    CNX$INIT_CSBS           ; Clear out locked, selected bits
            51    7C    0577          CLRQ    R1                      ; R1 is last CSB from a cluster node
                        0579                                          ; R2 is number of cluster nodes seen
          FA84'   30    0579          BSBW    CNX$SCAN_CSBS           ; Iterate over all known CSB's
         20 50    E9    057C          BLBC    R0,140$                 ; Branch when done
 1A 60 A3   00    E0    057F          BBS     #CSB$V_LONG_BREAK, -    ; Branch to skip if there has been
                        0584                  CSB$L_STATUS(R3), 130$  ;  a long break
 15 60 A3   19    E1    0584          BBC     #CSB$V_STATUS_RCVD, -   ; Branch if status not received
                        0589                  CSB$L_STATUS(R3), 130$
 07 60 A3   08    E0    0589          BBS     #CSB$V_CLUSTER, -       ; Branch if remote node is a cluster
```

```
                           058E                  CSB$L_STATUS(R3), 110$      ;  member
0B 60 A3    11       E3    058E           BBCS   #CSB$V_SELECTED, -          ; Mark node selected
                           0593                  CSB$L_STATUS(R3),130$
            09       11    0593           BRB    130$                        ; Join common exit
                           0595
            52       D6    0595  110$:    INCL   R2                          ; Count cluster member node
            51       D5    0597           TSTL   R1                          ; Seen a cluster node before?
            00       13    0599           BEQL   120$                        ; Branch if no
                           059B  ;
                           059B  ; Could check that R3 and R1 represent CSB's from the same cluster
                           059B  ;
      51    53       D0    059B  120$:    MOVL   R3,R1                       ; Update last seen cluster node
            05             059E  130$:    RSB
                           059F
            51       D5    059F  140$:    TSTL   R1                          ; Any cluster members seen?
            1F       12    05A1           BNEQ   150$                        ; Branch if members seen
                           05A3  ;
                           05A3  ; Consider whether it is feasible to try to form a cluster
                           05A3  ;
23 1C A4    12       E0    05A3           BBS    #CLUB$V_NO_FORM, -          ; Branch if cluster formation is
                           05A8                  CLUB$L_FLAGS(R4),160$
   53    10 A4       D0    05A8           MOVL   CLUB$L_LOCAL_CSB(R4),R3     ; Local CSB address
         50 A3       B5    05AC           TSTW   CSB$W_VOTES(R3)             ; If local node has no votes, don't
            1A       13    05AF           BEQL   160$                        ;   try to form cluster
         FA4C'       30    05B1           BSBW   CNX$QUORUM_CALC             ; Compute votes, quorum, nodes
      51    50       D1    05B4           CMPL   R0,R1                       ; Is there any possibility of a quorum?
            12       1F    05B7           BLSSU  160$                        ; No quorum possible, go to common exit
 00C8 C4    03       D3    05B9           BITL   #^B11, -                    ; Has a foreign cluster been seen recently?
                           05BE                  CLUB$L_FOREIGN_CLUSTER(R4)
            0B       12    05BE           BNEQ   160$                        ; Branch if yes and don't try to form a clus
            4D       11    05C0           BRB    FORM_CLUSTER                ; Try to form a cluster
                           05C2
   6E A1    52       B1    05C2  150$:    CMPW   R2,CSB$W_NODES(R1)          ; Do we perhaps see all members?
            03       1F    05C6           BLSSU  160$                        ; Nothing to do right now
          02BC       31    05C8           BRW    JOIN_CLUSTER                ; Try to join a cluster
                           05CB
          FEE8       31    05CB  160$:    BRW    START_TIMEOUT               ; Begin a timeout and then rescan
                           05CE
                     05    05CE  170$:    RSB                                ; Exit, dropping thread
                           05CF  ;
                           05CF  ; Here if a cluster member looking for something to do
                           05CF  ;
03 1C A4    17       E1    05CF  200$:    BBC    #CLUB$V_LOST_CNX, -         ; Branch if no cluster connection is broken
                           05D4                  CLUB$L_FLAGS(R4),220$
          018D       31    05D4           BRW    RECONFIG_CLUSTER            ; Need to do failover
                           05D7
      55    00       D2    05D7  220$:    MCOML  #0,R5                       ; Accumulate shutdown bits
         FA23'       30    05DA           BSBW   CNX$SCAN_CSBS               ; Iterate over all known CSB's
         0D 50       E9    05DD           BLBC   R0,240$                     ; Branch when done
07 60 A3    01       E1    05E0           BBC    #CSB$V_MEMBER, -            ; Branch if not a cluster member
                           05E5                  CSB$L_STATUS(R3), 230$
   50    60 A3       D2    05E5           MCOML  CSB$L_STATUS(R3),R0
      55    50       CA    05E9           BICL2  R0,R5                       ; Accumulate AND of all shutdown bits
                     05    05EC  230$:    RSB
                           05ED
   04 55    0A       E1    05ED  240$:    BBC    #CSB$V_SHUTDOWN,R5,250$     ; Branch if not ready to shutdown
                           05F1           BUG_CHECK      OPERATOR,FATAL      ; Cluster-wide shutdown
```

```
                          05F5  1566
     11 1C A4    1B   E0  05F5  1567 250$:    BBS     #CLUB$V_ADJ_QUORUM, -    ; Branch if quorum adjustment requested
                          05FA  1568                  CLUB$L_FLAGS(R4),260$
        10 55    03   E1  05FA  1569          BBC     #CSB$V_QF_SAME,R5,270$   ; Branch if not all members have same disk
       55   01   09   EF  05FE  1570          EXTZV   #CSB$V_QF_ACTIVE,#1, -   ; Get AND of all member QF active bits
                 50       0602  1571                  R5,R0                    ;    into R0
50   1C A4  01   19   ED  0603  1572          CMPZV   #CLUB$V_QF_VOTE,#1, -    ; Compare to present state of quorum disk
                          0609  1573                  CLUB$L_FLAGS(R4),R0
                 03   13  0609  1574          BEQL    270$                     ; Branch if present state is OK
               050B   31  060B  1575 260$:    BRW     ADJUST_QUORUM            ; Attempt to adjust quorum/quorum disk
                          060E  1576 ;
                          060E  1577 ; Nothing to do right now, so drop thread
                          060E  1578 ;
                      05  060E  1579 270$:    RSB
```

```
                            060F          .SBTTL  FORM_CLUSTER - Try to form a cluster
                            060F
                            060F   ;++
                            060F   ;
                            060F   ; FUNCTIONAL DESCRIPTION:
                            060F   ;
                            060F   ;        This routine tries to form a new cluster.  It assumes that no
                            060F   ;        cluster is known to this node.
                            060F   ;
                            060F   ; CALLING SEQUENCE:
                            060F   ;
                            060F   ;        JSB     FORM_CLUSTER
                            060F   ;        IPL is IPL$_SCS
                            060F   ;
                            060F   ; INPUT PARAMETERS:
                            060F   ;
                            060F   ;        NONE
                            060F   ;
                            060F   ; OUTPUT PARAMETERS:
                            060F   ;
                            060F   ;        NONE
                            060F   ;
                            060F   ; COMPLETION CODES:
                            060F   ;
                            060F   ;        NONE
                            060F   ;
                            060F   ; SIDE EFFECTS:
                            060F   ;
                            060F   ;        R0-R5 are destroyed
                            060F   ;
                            060F   ;--
                            060F
                            060F   FORM_CLUSTER:
             058F      30   060F           BSBW    INIT_TRANSITION             ; Initialization for becoming coordinator
           50 A3       B5   0612           TSTW    CSB$W_VOTES(R3)             ; If local node has no votes, don't
               6A      13   0615           BEQL    50$                         ;   try to form cluster
      58 A4    01      90   0617           MOVB    #CLMCNX$K_XTN_FORM, -       ; FORM transaction
                            061B                   CLUB$B_CUR_CODE(R4)
   2C A4    50 A4      7D   061B           MOVQ    CLUB$Q_CUR_TIME(R4), -      ; Define foundation time of cluster
                            0620                   CLUB$Q_FTIME(R4)
 00000018'GF   06      28   0620           MOVC3   #CLUB$S_FSYSID, -           ; Define founding system ID
           26 A4            0627                   G^SCS$GA_LOCALSB+SB$B_SYSTEMID, -
                            0629                   CLUB$B_FSYSID(R4)
               55      D4   0629           CLRL    R5                          ; No CSB for message
     50    0000'CF     9E   062B           MOVAB   TRYFORM_MSG,R0              ; Trying to form cluster message
            F9CD'      30   0630           BSBW    CNX$CONFIG_CHANGE           ; Make status known
            F9CA'      30   0633           BSBW    CNX$SCAN_CSBS               ; Examine all CSBs
            26 50      E9   0636           BLBC    R0,20$                      ; Branch when done
   14 60 A3    19      E1   0639           BBC     #CSB$V_STATUS_RCVD, -       ; Ignore nodes that haven't sent
                            063E                   CSB$L_STATUS(R3),10$        ;   status
   0F 60 A3    1A      E0   063E           BBS     #CSB$V_SEND_STATUS, -       ; Ignore nodes that need status
                            0643                   CSB$L_STATUS(R3),10$
   0A 60 A3    00      E0   0643           BBS     #CSB$V_LONG_BREAK, -        ; Ignore nodes with whom the connection
                            0648                   CSB$L_STATUS(R3),10$        ;   is broken
           50 A3       B5   0648           TSTW    CSB$W_VOTES(R3)             ; Ignore nodes with no votes
               05      13   064B           BEQL    10$
   00 60 A3    11      E3   064D           BBCS    #CSB$V_SELECTED, -          ; Mark node selected
```

```
                                    0652                   CSB$L_STATUS(R3),10$
                        53     DD   0652 10$:     PUSHL    R3                           ; Save context register
 008C C3   20   00   6E  00    2C   0654          MOVC5    #0,(SP),#0, -                ; Zero nodemap
                                    065C                   #CSB$S_NODEMAP, -
                                    065C                   CSB$B_NODEMAP(R3)
                        08     BA   065C          POPR     #^M<R3>
                               05   065E          RSB
                                    065F
                                    065F 20$:
                                    065F ;
                                    065F ; Selected nodes are the possible population from which to form a
                                    065F ; cluster.
                                    065F ;
                     F99E'     30   065F          BSBW     CNX$QUORUM_CALC              ; Compute votes, quorum, nodes
                  51    50     D1   0662          CMPL     R0,R1                        ; Is there any possibility of a quorum?
                        1A     1F   0665          BLSSU    50$                          ; No quorum possible
                      0ACD     30   0667          BSBW     LOCK_NODES                   ; Attempt to lock selected nodes
                                    066A ;
                                    066A ; Have locked all possible nodes.
                                    066A ; Assign temporary CSID's to nodes.
                                    066A ;
                     F993'     30   066A          BSBW     CNX$SCAN_CSBS
                     14 50     E9   066D          BLBC     R0,60$                       ; All done
            0B 60 A3    11     E1   0670          BBC      #CSB$V_SELECTED, -           ; Branch if not selected
                                    0675                   CSB$L_STATUS(R3),40$
                     F988'     30   0675          BSBW     CNX$ASSIGN_CSID              ; Create a CSID for the selected node
                     05 50     E8   0678          BLBS     R0,40$                       ; Branch on success
            00 1C A4    11     E3   067B          BBCS     #CLUB$V_UNLOCK, -            ; Request that everything be unlocked
                                    0680                   CLUB$L_FLAGS(R4), 40$
                               05   0680 40$:     RSB                                   ; Resume scan
                                    0681
                      0BC2     31   0681 50$:     BRW      UNLOCK_ALL                   ; Bail out of transaction
                                    0684
                      0BAD     30   0684 60$:     BSBW     CNX$CHECK_UNLOCK             ; Check for unlock request
                                    0687 ;
                                    0687 ; Prepare to describe all candidate nodes to all other candidate nodes
                                    0687 ;
                                    0687 FORM_DESCRIBE:
                     5A A4     B4   0687          CLRW     CLUB$W_MSGCNT(R4)            ; Count of responses to wait for
                     F973'     30   068A          BSBW     CNX$SCAN_CSBS                ; Scan all CSBs
                     76 50     E9   068D          BLBC     R0,130$                      ; Branch if scan done
            31 60 A3    11     E1   0690          BBC      #CSB$V_SELECTED, -           ; Branch if not selected
                                    0695                   CSB$L_STATUS(R3),10$
                51    10 A4    D0   0695          MOVL     CLUB$L_LOCAL_CSB(R4),R1      ; Local CSB address
                50    4C A3    3C   0699          MOVZWL   CSB$W_CSID_IDX(R3),R0        ; Subject node CSID index
         00 008C C1     50     E2   069D          BBSS     R0,CSB$B_NODEMAP(R1),5$      ; Mark local node able to see node
            1E 60 A3    00     E0   06A3 5$:      BBS      #CSB$V_LONG_BREAK, -         ; Branch if permanent break in
                                    06A8                   CSB$L_STATUS(R3),10$         ;   connection
            19 60 A3    18     E0   06A8          BBS      #CSB$V_LOCAL, -              ; Branch if local CSB
                                    06AD                   CSB$L_STATUS(R3),10$
                     F950'     30   06AD          BSBW     CNX$ALLOC_CDRP_ONLY          ; Get a fork block
                     F94D'     30   06B0          BSBW     CNX$RESOURCE_CHECK           ; Watch out for exhausted resources
                     11 50     E9   06B3          BLBC     R0,20$                       ; No memory available
                  48 A5 53     D0   06B6          MOVL     R3,CDRP$L_VAL8(R5)           ; Address of node to talk to
                  44 A5 64     7E   06BA          MOVAQ    CLUB$L_CSBQFL(R4), -         ; Address of CSB listhead
                                    06BE                   CDRP$L_VAL7(R5)
                     5A A4     B6   06BE          INCW     CLUB$W_MSGCNT(R4)            ; Increment waiting count
```

```
            07   10  06C1       BSBB    40$                       ; Describe nodes
         F93A'   31  06C3       BRW     CNX$SCAN_CSBS_FORK        ; Fork and continue scan
                     06C6
            05       06C6 10$:  RSB
                     06C7
         F936'   31  06C7 20$:  BRW     CNX$SCAN_CSBS_RETRY       ; Wait and resume scan
                     06CA
                     06CA ;
                     06CA ; Tell a node about all of the others
                     06CA ;
      52  44 B5  D0  06CA 40$:  MOVL    @CDRP$L_VAL7(R5),R2       ; Next CSB to describe
      44 A5  52  D0  06CE       MOVL    R2,CDRP$L_VAL7(R5)        ;  and save it
                     06D2       ASSUME  CLUB$L_CSBQFL EQ 0
         54  52  D1  06D2       CMPL    R2,R4                     ; At end of list?
         26  13      06D5       BEQL    125$                      ; Branch if done
   EE 60 A2  11  E1  06D7       BBC     #CSB$V_SELECTED, -        ; Branch if not selected
                     06DC               CSB$L_STATUS(R2),40$
         086C    30  06DC       BSBW    DESCRIBE_NODE             ; Tell one node about another
      54  64 A3  D0  06DF       MOVL    CSB$L_CLUB(R3),R4         ; Address of CLUB
         12 50   E9  06E3       BLBC    R0,120$                   ; Error sending message
         0A 51   E9  06E6       BLBC    R1,50$                    ; Branch if node is unknown
      50  4C A2  3C  06E9       MOVZWL  CSB$W_CSID_IDX(R2),R0     ; CSID Index of described system
00 008C C3   50  E2  06ED       BBSS    R0,CSB$B_NODEMAP(R3), -   ; Mark node visible
                     06F3               50$
   D2 1C A4  11  E1  06F3 50$:  BBC     #CLUB$V_UNLOCK, -         ; Branch if no unlock request
                     06F8               CLUB$L_FLAGS(R4),40$
                     06F8 ;
   00 1C A4  11  E2  06F8 120$: BBSS    #CLUB$V_UNLOCK, -         ; Request that all be unlocked
                     06FD               CLUB$L_FLAGS(R4),125$
         50  55  D0  06FD 125$: MOVL    R5,R0                     ; Address of CDRP
     00000000'GF 16  0700       JSB     G^EXE$DEANONPAGED         ; Deallocate CDRP
                     0706
                     0706 ;
                     0706 ; terminate threads -- except for the last one
                     0706 ;
         5A A4   B7  0706 130$: DECW    CLUB$W_MSGCNT(R4)
            01   19  0709       BLSS    140$                      ; Branch when done
            05       070B       RSB                               ; Terminate thread
                     070C
         0B25    30  070C 140$: BSBW    CNX$CHECK_UNLOCK          ; Check unlock request
                     070F
                     070F ;
                     070F ; Get here when all nodes have been described to all others
                     070F ; Now have told every node about every other node and each node's
                     070F ; CSB contains a map of nodes that can be seen
                     070F ;
                     070F FORM_PROPOSE:
         F8EE'   30  070F       BSBW    CNX$OPT_INIT              ; Compute optimal cluster containing local n
         23 50   E9  0712       BLBC    R0,270$                   ; Can't perform computation (no pool?)
         F8E8'   30  0715       BSBW    CNX$SCAN_CSBS             ; Scan all CSBs
         15 50   E9  0718       BLBC    R0,260$                   ; Branch when scan is done
   0F 60 A3  11  E1  071B       BBC     #CSB$V_SELECTED, -        ; Branch if CSB not selected
                     0720               CSB$L_STATUS(R3),250$
      50  4C A3  3C  0720       MOVZWL  CSB$W_CSID_IDX(R3),R0     ; CSID index of CSB
05 00EC C4   50  E0  0724       BBS     R0, -                     ; Branch if node is in desired cluster
                     072A               CLUB$B_NODEMAP(R4),250$ ;
   00 60 A3  11  E5  072A       BBCC    #CSB$V_SELECTED, -        ; Clear select bit
```

```
                              072F            CSB$L_STATUS(R3),250$
                         05   072F  250$:  RSB                                ; Advance to next iteration
                              0730
                              0730  ;
                              0730  ; Check for quorum
                              0730  ;
                    F8CD'  30  0730  260$:  BSBW   CNX$QUORUM_CALC             ; Compute quorum, votes, nodes
                 51    50  D1  0733         CMPL   R0,R1                       ; Is there any possibility of a quorum?
                       03  1E  0736         BGEQU  290$                        ; Have a quorum!
                     0B0B  31  0738  270$:  BRW    UNLOCK_ALL                  ; Bail out of transition
                               073B
                    F8C2'  30  073B  290$:  BSBW   CNX$DIRVEC_ADJ              ; Create directory system vector
                    F7 50  E9  073E         BLBC   R0,270$                     ; Branch on failure
    0094 C4    00000000'GF  7D  0741         MOVQ   G^EXE$GQ_SYSTIME, -         ; Save current time as new
                               074A                CLUB$Q_NEWTIME(R4)
       009C C4    0094 C4  7D  074A         MOVQ   CLUB$Q_NEWTIME(R4), -       ; Save as reference time also
                               0751                CLUB$Q_NEWTIME_REF(R4)
                               0751  ;
                               0751  ; Propose cluster -- Send Phase 1 Messages
                               0751  ;
            70 A4       07  9A  0751         MOVZBL #CLMCNX$K_FNC_FORM, -       ; Facility specific message code
                               0755                CLUB$L_CTX0(R4)
         78 A4    0BFA'CF  9E  0755         MOVAB  W^BLD_FORM_MSG, -           ; Address of routine to build
                               075B                CLUB$L_CTXT(R4)             ;  status message
                     0901  30  075B         BSBW   SEND_PH1                    ; Send phase 1 messages
                               075E  ;
                               075E  ; All nodes have seen Phase 1 and ACKed it.
                               075E  ; Now is the time to send Phase 2.
                               075E  ;
                     095D  31  075E         BRW    SEND_PH2
                               0761  ;
                               0761  ; At the completion of sending Phase 2 messages, this location
                               0761  ; is executed as a thread.
                               0761  ;
                               0761  FORM_FINISH:
                     0322  31  0761         BRW    FORM_JOIN_FINISH            ; Branch to common form/join phase 2 process
```

```
                        0764          .SBTTL  RECONFIG_CLUSTER - Reconfigure a cluster following a node failure
                        0764
                        0764    ;++
                        0764    ;
                        0764    ; FUNCTIONAL DESCRIPTION:
                        0764    ;
                        0764    ;       This routine tries to reconfigure a cluster following a connection failure.
                        0764    ;       This implements a fast and simple scheme that works only if a failure involv
                        0764    ;       all connections to a failed set of nodes.  A second scheme is necessary to
                        0764    ;       deal with the more complicated case of random connection failures.
                        0764    ;
                        0764    ; CALLING SEQUENCE:
                        0764    ;
                        0764    ;       JSB     RECONFIG_CLUSTER
                        0764    ;       IPL is IPL$_SCS
                        0764    ;
                        0764    ; INPUT PARAMETERS:
                        0764    ;
                        0764    ;       NONE
                        0764    ;
                        0764    ; OUTPUT PARAMETERS:
                        0764    ;
                        0764    ;       NONE
                        0764    ;
                        0764    ; COMPLETION CODES:
                        0764    ;
                        0764    ;       NONE
                        0764    ;
                        0764    ; SIDE EFFECTS:
                        0764    ;
                        0764    ;       R0-R5 are destroyed
                        0764    ;
                        0764    ;--
                        0764
                        0764    RECONFIG_CLUSTER:
          043A    30    0764          BSBW    INIT_TRANSITION            ; Initialization for becoming coordinator
    58 A4   03    90    0767          MOVB    #CLM$CNX$K_XTN_RECONFIG, - ; RECONFIGure transaction
                        076B                  CLUB$B_CUR_CODE(R4)
            55    D4    076B          CLRL    R5                         ; No CSB address
50   0000'CF    9E    076D          MOVAB   RECONFIG_MSG,R0            ; Initiating cluster reconfiguration message
         F88B'    30    0772          BSBW    CNX$CONFIG_CHANGE          ; Make this state known
         F888'    30    0775          BSBW    CNX$SCAN_CSBS
         10 50    E9    0778          BLBC    R0,30$                     ; Branch when done
 0A 60 A3   01    E1    077B          BBC     #CSB$V_MEMBER, -           ; Ignore nodes that are not members of
                        0780                  CSB$L_STATUS(R3),10$       ;   the local cluster
 05 60 A3   00    E0    0780          BBS     #CSB$V_LONG_BREAK, -       ; Ignore nodes with whom the connection
                        0785                  CSB$L_STATUS(R3),10$       ;   is irrevocably broken
 00 60 A3   11    E3    0785          BBCS    #CSB$V_SELECTED, -         ; Mark node selected
                        078A                  CSB$L_STATUS(R3),10$
            05          078A    10$:  RSB
                        078B
                        078B    30$:
                        078B    ;
                        078B    ; Selected nodes are those proposed for the members of the reconfigured
                        078B    ; cluster.
                        078B    ;
          09A9    30    078B          BSBW    LOCK_NODES                 ; Attempt to lock selected nodes
```

```
                          078E
                          078E  RECONFIG_LOCKED:
                          078E  ;
                          078E  ; Now, exchange topology messages with all member nodes
                          078E  ;
         5A A4      B4    078E          CLRW    CLUB$W_MSGCNT(R4)           ; Initialize waiting count
      59 A4    30   90    0791          MOVB    #CLMCNX$K_DATA, -           ; Mark this as Phase 1
                          0795                  CLUB$B_CUR_PHASE(R4)
         F868'     30     0795          BSBW    CNX$SCAN_CSBS               ; Iterate over all CSBs
         44 50     E9     0798          BLBC    R0,90$                      ; Branch when done
   1B 60 A3    11  E1     079B          BBC     #CSB$V_SELECTED, -          ; Branch if not selected
                          07A0                  CSB$L_STATUS(R3),40$
   16 60 A3    18  E0     07A0          BBS     #CSB$V_LOCAL, -             ; Branch if local CSB
                          07A5                  CSB$L_STATUS(R3),40$
   11 60 A3    00  E0     07A5          BBS     #CSB$V_LONG_BREAK, -        ; Branch if connection is
                          07AA                  CSB$L_STATUS(R3),40$        ;   permanently broken
         F853'     30     07AA          BSBW    CNX$ALLOC_WARMCDRP_CSB
         F850'     30     07AD          BSBW    CNX$RESOURCE_CHECK          ; Watch out for exhausted resources
         09 50     E9     07B0          BLBC    R0,50$                      ; No memory available
         5A A4     B6     07B3          INCW    CLUB$W_MSGCNT(R4)           ; Include in wait count
            07     10     07B6          BSBB    60$                         ; Describe nodes
         F845'     31     07B8          BRW     CNX$SCAN_CSBS_FORK          ; Fork and continue scan
                          07BB
                   05     07BB  40$:    RSB                                 ; Return and continue scan
                          07BC
         F841'     31     07BC  50$:    BRW     CNX$SCAN_CSBS_RETRY         ; Delay and resume scan
                          07BF
                          07BF  ;
                          07BF  ; Send topology message to one node
                          07BF  ;
      50    0F     9A     07BF  60$:    MOVZBL  #CLMCNX$K_FNC_TOPOLOGY,R0 ; Message code
          0B4D     30     07C2          BSBW    INIT_STD_MSG                ; Standard CDRP message initialization
   4C A5  1022'CF  9E     07C5          MOVAB   W^BLD_TOPOLOGY_MSG, -       ; Address of routine to build
                          07CB                  CDRP$L_MSGBLD(R5)           ;  topology message
         F832'     30     07CB          BSBW    CNX$SEND_MSG_CSB            ; Send message
                          07CE  ;
                          07CE  ;       We are resumed here when the response message arrives.
                          07CE  ;       Registers contain:
                          07CE  ;               R0:     Status
                          07CE  ;               R2:     Address of message buffer
                          07CE  ;               R3:     Address of CSB
                          07CE  ;               R4:     Address of PDT
                          07CE  ;               R5:     Address of CDRP
                          07CE  ;
         0B 50     E9     07CE          BLBC    R0,80$                      ; Branch on no data received
            3D     BB     07D1          PUSHR   #^M<R0,R2,R3,R4,R5>         ; Save registers
008C C3   14 A2   20   28 07D3          MOVC3   #CSB$S_NODEMAP, -           ; Fill in connectivity in sending node's
                          07DA                  CLMTOP$B_NODEMAP(R2), -     ;   CSB
                          07DA                  CSB$B_NODEMAP(R3)
            3D     BA     07DA          POPR    #^M<R0,R2,R3,R4,R5>         ; Restore registers
          0B07     30     07DC  80$:    BSBW    CNX$PROCESS_RESPONSE        ; Deallocate storage
                          07DF  ;
                          07DF  ; terminate threads -- except for the last one
                          07DF  ;
         5A A4     B7     07DF  90$:    DECW    CLUB$W_MSGCNT(R4)
            01     19     07E2          BLSS    100$                        ; Branch when done
                   05     07E4          RSB                                 ; Terminate thread
```

```
                                 07E5
                  0A4C   30      07E5  100$:    BSBW    CNX$CHECK_UNLOCK            ; Handle a pending unlock request
                                 07E8
                                 07E8  RECONFIG_PROPOSE:
                                 07E8  ;
                                 07E8  ; Get here when all nodes in the proposed new cluster have been locked.
                                 07E8  ; Compute and send Phase 1 proposal messages.
                                 07E8  ;
                 F815'   30      07E8           BSBW    CNX$OPT_INIT                ; Compute optimal subcluster
                21 50    E9      07EB           BLBC    R0,25$                      ; Can't compute (no pool?)
                 F80F'   30      07EE           BSBW    CNX$SCAN_CSBS               ; Scan all CSB's
                15 50    E9      07F1           BLBC    R0,20$                      ; Branch when done
      0F 60 A3     11    E1      07F4           BBC     #CSB$V_SELECTED, -          ; Branch if not selected
                                 07F9                   CSB$L_STATUS(R3),10$
        50   4C A3       3C      07F9           MOVZWL  CSB$W_CSID_IDX(R3),R0       ; CSID index of CSB
   05 00EC C4      50    E0      07FD           BBS     R0, -                       ; Node is in computed subcluster
                                 0803                   CLUB$B_NODEMAP(R4),10$      ;
      00 60 A3     11    E5      0803           BBCC    #CSB$V_SELECTED, -          ; De-select node not in computed cluster
                                 0808                   CSB$L_STATUS(R3),10$
                         05      0808  10$:     RSB                                 ; Return and resume scan
                                 0809
                                 0809  20$:
                 F7F4'   30      0809           BSBW    CNX$DIRVEC_ADJ              ; Adjust size of lock manager directory vect
                03 50    E8      080C           BLBS    R0,30$                      ; Branch on success
                  0A34   31      080F  25$:     BRW     UNLOCK_ALL
                                 0812
                 F7EB'   30      0812  30$:     BSBW    CNX$QUORUM_CALC             ; Calculate and store quorum related paramet
                                 0815  ;
                                 0815  ; Propose cluster -- Send Phase 1 Messages
                                 0815  ;
         70 A4     08    9A      0815           MOVZBL  #CLMCNX$K_FNC_RECONFIG, - ; Message code
                                 0819                   CLUB$L_CTX0(R4)
   78 A4   0BFA'CF       9E      0819           MOVAB   W^BLD_RECONFIG_MSG, -       ; Address of routine to build
                                 081F                   CLUB$L_CTX1(R4)             ;  status message
                  083D   30      081F           BSBW    SEND_PH1                    ; Send phase 1 messages
                                 0822  ;
                                 0822  ; All nodes have seen Phase 1 and ACKed it.
                                 0822  ; Now is the time to send Phase 2.
                                 0822  ;
                  0899   31      0822           BRW     SEND_PH2
                                 0825  ;
                                 0825  ; At the completion of sending Phase 2 messages, this location
                                 0825  ; is executed as a thread.
                                 0825  ;
                                 0825  RECONFIG_FINISH:
                 F7D8'   30      0825           BSBW    CNX$SCAN_CSBS               ; Iterate over all CSBs
                18 50    E9      0828           BLBC    R0,30$                      ; Branch when scan is done
      12 60 A3     11    E1      082B           BBC     #CSB$V_SELECTED, -          ; Branch if node not selected
                                 0830                   CSB$L_STATUS(R3),20$
            50     1F    D0      0830           MOVL    #CSB$S_NODEMAP-1,R0         ; Byte count of nodemap
   52   00EC C440        92      0833  10$:     MCOMB   CLUB$B_NODEMAP(R4)[R0],R2
   008C C340       52    8A      0839           BICB2   R2,CSB$B_NODEMAP(R3)[R0] ; CSB = CSB & CLUB
                F1 50    F4      083F           SOBGEQ  R0,10$
                         05      0842  20$:     RSB
                                 0843
                  0B1D   30      0843  30$:     BSBW    UPDATE_QUORUM               ; Update quorum, votes, nodes
                                 0846  ;
```

```
                          0846  ; Handle bookkeeping for removed nodes.
                          0846  ;
              F7B7'  30   0846          BSBW    CNX$SCAN_CSBS            ; Iterate over all CSBs
              13 50  E9   0849          BLBC    R0,60$                   ; Branch when done
     0A 60 A3   11   E0   084C          BBS     #CSB$V_SELECTED, -       ; Branch if CSB selected -
                          0851                  CSB$L_STATUS(R3),40$     ;    still a cluster member
     08 60 A3   01   E1   0851          BBC     #CSB$V_MEMBER, -         ; Branch if not a cluster member
                          0856                  CSB$L_STATUS(R3),50$
              0BA0   30   0856          BSBW    REMOVE_NODE              ; Mark the node out of the cluster
                03   11   0859          BRB     50$                      ; Branch to common exit
                          085B
              F7A2'  30   085B  40$:    BSBW    CNX$MARK_UNLOCKED        ; Mark node unlocked
                     05   085E  50$:    RSB                              ; Continue scan
                          085F
                          085F  60$:
                          085F  ;
                          085F  ; Recompute value of CLUB$V_LOST_CNX
                          085F  ;
     00 1C A4   17   E5   085F          BBCC    #CLUB$V_LOST_CNX, -      ; Clear bit prior to recomputing its
                          0864                  CLUB$L_FLAGS(R4),70$
              F799'  30   0864  70$:    BSBW    CNX$SCAN_CSBS            ; Iterate over CSBs
              10 50  E9   0867          BLBC    R0,220$                  ; Branch when scan is done
     0A 60 A3   01   E1   086A          BBC     #CSB$V_MEMBER, -         ; Branch if node is not a cluster member
                          086F                  CSB$L_STATUS(R3), 210$
     05 60 A3   00   E1   086F          BBC     #CSB$V_LONG_BREAK, -     ; Branch if no long break has happened
                          0874                  CSB$L_STATUS(R3),210$
     00 1C A4   17   E2   0874          BBSS    #CLUB$V_LOST_CNX, -      ; Mark cluster connection lost
                          0879                  CLUB$L_FLAGS(R4),210$
                     05   0879  210$:   RSB
                          087A
              F783'  30   087A  220$:   BSBW    CNX$DIRVEC_FILL          ; Recompute lock manager directory vector
           00AC C4   B6   087D          INCW    CLUB$W_MEMSEQ(R4)        ; Advance membership state sequence number
              F77C'  30   0881          BSBW    CNX$MEMBERSHIP_CHANGE    ; Begin table processing to remove nodes
              09BF   31   0884          BRW     UNLOCK_ALL               ; Make sure everything is unlocked
```

```
                    0887           .SBTTL  JOIN_CLUSTER - Try to join a cluster
                    0887
                    0887   ;++
                    0887   ;
                    0887   ; FUNCTIONAL DESCRIPTION:
                    0887   ;
                    0887   ;       This routine is called by a newly booted node that decides to
                    0887   ;       request membership in a cluster.
                    0887   ;
                    0887   ; CALLING SEQUENCE:
                    0887   ;
                    0887   ;       JSB     JOIN_CLUSTER
                    0887   ;       IPL is IPL$_SCS
                    0887   ;
                    0887   ; INPUT PARAMETERS:
                    0887   ;
                    0887   ;       NONE
                    0887   ;
                    0887   ; OUTPUT PARAMETERS:
                    0887   ;
                    0887   ;       NONE
                    0887   ;
                    0887   ; COMPLETION CODES:
                    0887   ;
                    0887   ;       NONE
                    0887   ;
                    0887   ; SIDE EFFECTS:
                    0887   ;
                    0887   ;       R0-R5 are destroyed
                    0887   ;
                    0887   ;--
                    0887
                    0887   JOIN_CLUSTER:
          55  D4    0887           CLRL    R5                          ; Will hold CSB address on return
       F774'  30    0889           BSBW    CNX$SCAN_CSBS               ; Examine all CSBs
       16 50  E9    088C           BLBC    R0,20$                      ; Branch when done
10 60 A3  19  E1    088F           BBC     #CSB$V_STATUS_RCVD, -       ; Ignore nodes that haven't sent
                    0894                   CSB$L_STATUS(R3),10$        ;   status
0B 60 A3  08  E1    0894           BBC     #CSB$V_CLUSTER, -           ; Ignore non-cluster members
                    0899                   CSB$L_STATUS(R3),10$
06 60 A3  00  E0    0899           BBS     #CSB$V_LONG_BREAK, -        ; Ignore nodes with whom the connection
                    089E                   CSB$L_STATUS(R3),10$        ;   is broken
       55 53  D0    089E           MOVL    R3,R5
       F75C'  31    08A1           BRW     CNX$SCAN_CSBS_EXIT          ; Return CSB address of cluster member
                    08A4
              05    08A4   10$:    RSB
                    08A5
       53 55  D0    08A5   20$:    MOVL    R5,R3                       ; CSB address of node to apply to
          2C  13    08A8           BEQL    50$                         ; No node to apply to at this time
27 60 A3  0A  E0    08AA           BBS     #CSB$V_SHUTDOWN, -          ; Branch if node in shutdown mode
                    08AF                   CSB$L_STATUS(R3),50$
22 60 A3  1A  E0    08AF           BBS     #CSB$V_SEND_STATUS, -       ; Branch if status not yet queued
                    08B4                   CSB$L_STATUS(R3),50$        ;   for transmission
       F749'  30    08B4           BSBW    CNX$ALLOC_CDRP_ONLY         ; Get a CDRP
       1C 50  E9    08B7           BLBC    R0,50$                      ; No memory available
   54 64 A3   D0    08BA           MOVL    CSB$L_CLUB(R3),R4           ; CLUB address
58 A4     02  90    08BE           MOVB    #CLMCNX$K_XTN_JOIN, -       ; Transition identifier
```

```
                    08C2  2052                      CLUB$B_CUR_CODE(R4)
     50     02  9A  08C2  2053          MOVZBL  #CLMCNX$K_FNC_ENTER,R0   ; Facility specific message code
          0A4A  30  08C5  2054          BSBW    INIT_STD_MSG             ; Standard CDRP message initialization
         F735'  30  08C8  2055          BSBW    CNX$SEND_FORGET          ; Send message and forget it
     55     53  D0  08CB  2056          MOVL    R3,R5                    ; Address of CSB
  50   0000'CF  9E  08CE  2057          MOVAB   REQJOIN_MSG,R0           ; Requesting cluster membership
         F72A'  30  08D3  2058          BSBW    CNX$CONFIG_CHANGE        ; Make this state known
          FBD6  31  08D6  2059 50$:     BRW     START_LONG_TIMEOUT
```

```
                             08D9          .SBTTL  CNX$RCVD_ENTER - Received a cluster membership request
                             08D9
                             08D9  ;++
                             08D9  ;
                             08D9  ; FUNCTIONAL DESCRIPTION:
                             08D9  ;
                             08D9  ;       This routine is called when a cluster membership request is received.
                             08D9  ;       If the request looks valid, the protocol for adding a node to a cluster
                             08D9  ;       is executed.
                             08D9  ;
                             08D9  ; CALLING SEQUENCE:
                             08D9  ;
                             08D9  ;       JSB     CNX$RCVD_ENTER
                             08D9  ;       IPL is IPL$_SCS
                             08D9  ;
                             08D9  ; INPUT PARAMETERS:
                             08D9  ;
                             08D9  ;       R2:     Message address
                             08D9  ;       R3:     CSB of sending system
                             08D9  ;       R4:     PDT address
                             08D9  ;
                             08D9  ; OUTPUT PARAMETERS:
                             08D9  ;
                             08D9  ;       NONE
                             08D9  ;
                             08D9  ; COMPLETION CODES:
                             08D9  ;
                             08D9  ;       NONE
                             08D9  ;
                             08D9  ; SIDE EFFECTS:
                             08D9  ;
                             08D9  ;       R0-R5 are destroyed
                             08D9  ;
                             08D9  ;--
                             08D9
                             08D9  CNX$RCVD_ENTER::
               F724'   30    08D9          BSBW    CNX$DEALL_MSG_BUF_CSB   ; Deallocate the message buffer
            55    53   D0    08DC          MOVL    R3,R5                   ; Address of CSB
      50   0000'CF     9E    08DF          MOVAB   MEMREQ_MSG,R0           ; Address of message block
               F719'   30    08E4          BSBW    CNX$CONFIG_CHANGE       ; Received cluster membership request
         54   64 A3    D0    08E7          MOVL    CSB$L_CLUB(R3),R4       ; Address of CLUB
1C A4    20800004 8F   D3    08EB          BITL    #< -                    ; Check for:
                             08F3                  CLUB$M_TRANSITION ! - ;   Transition in progress
                             08F3                  CLUB$M_LOST_CNX ! -   ;   Node addition inhibited
                             08F3                  CLUB$M_SHUTDOWN>, -   ;   Shutdown in progress
                             08F3                  CLUB$L_FLAGS(R4)
               30      12    08F3          BNEQ    20$                     ; Branch to ignore request
      2B 1C A4    00   E1    08F5          BBC     #CLUB$V_CLUSTER, -      ; Branch if this node is not a cluster
                             08FA                  CLUB$L_FLAGS(R4), 20$  ;   member
60 A3    04000007 8F   D3    08FA          BITL    #< -                    ; Check for:
                             0902                  CSB$M_SEND_STATUS ! - ;   Requestor is out of date
                             0902                  CSB$M_MEMBER ! -      ;   Requestor already a cluster member
                             0902                  CSB$M_LONG_BREAK ! -  ;   Connection problem to requestor *** why?
                             0902                  CSB$M_REMOVED>, -     ;   Requestor removed from cluster
                             0902                  CSB$L_STATUS(R3)
               21      12    0902          BNEQ    20$                     ; Branch to ignore request
   1C 60 A3    19      E1    0904          BBC     #CSB$V_STATUS_RCVD, -   ; Branch if we don't have info on
```

```
                              0909                    CSB$L_STATUS(R3), 20$     ;   the requestor
         70 A4    53    D0    0909            MOVL    R3,CLUB$L_CTXO(R4)        ; Save joining node CSB address
                              090D    ;
                              090D    ; The following piece of code assumes that CNX$SCAN_CSBS returns CSBs in order of
                              090D    ; increasing system ID.
                              090D    ;
            55    01    D0    090D            MOVL    #1,R5                     ; Assume local node has lowest System ID
              F6ED'     30    0910            BSBW    CNX$SCAN_CSBS             ; Iterate over all CSBs
            10    50    E9    0913            BLBC    R0,30$                    ; Must have found something
      0A 60 A3    01    E1    0916            BBC     #CSB$V_MEMBER, -          ; Branch if CSB is not for a member
                              091B                    CSB$L_STATUS(R3),20$
      02 60 A3    18    E0    091B            BBS     #CSB$V_LOCAL, -           ; Branch if not local CSB
                              0920                    CSB$L_STATUS(R3),10$
                  55    D4    0920            CLRL    R5                        ; Local node is not lowest system ID
              F6DB'     31    0922    10$:    BRW     CNX$SCAN_CSBS_EXIT
                              0925
                        05    0925    20$:    RSB
                              0926
            FC    55    E9    0926    30$:    BLBC    R5,20$                    ; If this is not the right node, drop thread
                              0929    ;
                              0929    ; Here, the transition to add the node begins
                              0929    ;
         58 A4    02    90    0929            MOVB    #CLMCNX$K_XTN_JOIN, -     ; JOIN transaction
                              092D                    CLUB$B_CUR_CODE(R4)
              0271      30    092D            BSBW    INIT_TRANSITION           ; Initialization for becoming coordinator
      55    70 A4       D0    0930            MOVL    CLUB$L_CTXO(R4),R5        ; CSB of joining node
   50    0000'CF        9E    0934            MOVAB   JOIN_MSG,R0               ; Initiating addition of system to cluster
              F6C4'     30    0939            BSBW    CNX$CONFIG_CHANGE         ; Make this state known
              F6C1'     30    093C            BSBW    CNX$SCAN_CSBS             ; Iterate over all CSBs
            0B    50    E9    093F            BLBC    R0, 60$                   ; Branch when done
      05 60 A3    01    E1    0942            BBC     #CSB$V_MEMBER, -          ; Branch if not a cluster member
                              0947                    CSB$L_STATUS(R3), 50$
      00 60 A3    11    E2    0947            BBSS    #CSB$V_SELECTED, -        ; Select member nodes
                              094C                    CSB$L_STATUS(R3), 50$
                        05    094C    50$:    RSB
                              094D
      53    70 A4       D0    094D    60$:    MOVL    CLUB$L_CTXO(R4),R3        ; CSB address of joining node
60 A3    00020000 8F    C8    0951            BISL2   #CSB$M_SELECTED, -        ; Mark joining node selected
                              0959                    CSB$L_STATUS(R3)
              F6A4'     30    0959            BSBW    CNX$QUORUM_CALC           ; Calculate quorum related parameters
            51    50    D1    095C            CMPL    R0,R1                     ; Votes >= quorum?
                  0A    1E    095F            BGEQU   80$                       ; Branch if yes
      20 A4    22 A4    B1    0961            CMPW    CLUB$W_VOTES(R4), -       ; Is there currently a quorum present?
                              0966                    CLUB$W_QUORUM(R4)
                  03    1F    0966            BLSSU   80$                       ; Branch if yes, don't undo quorum by adding
              08DB      31    0968    70$:    BRW     UNLOCK_ALL                ; Abort transition
                              096B
              F692'     30    096B    80$:    BSBW    CNX$ASSIGN_CSID           ; Assign a CSID
            F7    50    E9    096E            BLBC    R0,70$                    ; Branch on failure
                              0971
              07C3      30    0971            BSBW    LOCK_NODES                ; Attempt to lock selected nodes
                              0974
                              0974    JOIN_LOCKED:
                              0974
   54    00000000'GF    D0    0974            MOVL    G^CLU$GL_CLUB,R4          ; Address of CLUB
            5A A4       B4    097B            CLRW    CLUB$W_MSGCNT(R4)         ; Count of responses to wait for
                              097E    ;
```

```
                       097E   ; Describe all member nodes to joining system
                       097E   ;
          F67F'   30   097E            BSBW    CNX$SCAN_CSBS              ; Iterate over all CSBs
           2D 50  E9   0981            BLBC    R0,30$                     ; Branch when all systems described
  21 60 A3   11   E1   0984            BBC     #CSB$V_SELECTED, -         ; Branch if node not selected
                       0989                    CSB$L_STATUS(R3),10$
  1D 1C A4   11   E0   0989            BBS     #CLUB$V_UNLOCK, -          ; Branch if unlock requested
                       098E                    CLUB$L_FLAGS(R4),15$       ;    and abandon this transition
          F66F'   30   098E            BSBW    CNX$ALLOC_CDRP_ONLY        ; Get a fork block
          F66C'   30   0991            BSBW    CNX$RESOURCE_CHECK         ; Watch out for exhausted resources
           17 50  E9   0994            BLBC    R0,20$                     ; No memory available
              53  DD   0997            PUSHL   R3                         ; Save scan context
           52 53  D0   0999            MOVL    R3,R2                      ; CSB of node to describe
      53   70 A4  D0   099C            MOVL    CLUB$L_CTX0(R4),R3         ; Address of joining node CSB
           5A A4  B6   09A0            INCW    CLUB$W_MSGCNT(R4)          ; Increment waiting count
              3A  10   09A3            BSBB    60$                        ; Provide immediate return address
              08  BA   09A5            POPR    #^M<R3>                    ; Restore scanning context
          F656'   31   09A7            BRW     CNX$SCAN_CSBS_FORK         ; Fork and resume scan
                       09AA
                  05   09AA   10$:     RSB
                       09AB
          F652'   31   09AB   15$:     BRW     CNX$SCAN_CSBS_EXIT         ; Exit from iteration
                       09AE
          F64F'   31   09AE   20$:     BRW     CNX$SCAN_CSBS_RETRY        ; Wait and resume scan
                       09B1
                       09B1   ;
                       09B1   ; Describe joining system to all member systems
                       09B1   ;
          F64C'   30   09B1   30$:     BSBW    CNX$SCAN_CSBS              ; Scan all CSBs
           43 50  E9   09B4            BLBC    R0,130$                    ; Branch if scan done
  1F 60 A3   11   E1   09B7            BBC     #CSB$V_SELECTED, -         ; Branch if not selected
                       09BC                    CSB$L_STATUS(R3),40$
  1A 60 A3   18   E0   09BC            BBS     #CSB$V_LOCAL, -            ; Branch if local CSB
                       09C1                    CSB$L_STATUS(R3),40$
  15 60 A3   00   E0   09C1            BBS     #CSB$V_LONG_BREAK, -       ; Branch if connection permanently
                       09C6                    CSB$L_STATUS(R3),40$       ;    broken
          F637'   30   09C6            BSBW    CNX$ALLOC_CDRP_ONLY        ; Get a fork block
          F634'   30   09C9            BSBW    CNX$RESOURCE_CHECK         ; Watch out for exhausted resources
           0D 50  E9   09CC            BLBC    R0,50$                     ; No memory available
      52   70 A4  D0   09CF            MOVL    CLUB$L_CTX0(R4),R2         ; Address of joining node CSB
           5A A4  B6   09D3            INCW    CLUB$W_MSGCNT(R4)          ; Increment waiting count
              07  10   09D6            BSBB    60$                        ; Describe nodes
          F625'   31   09D8            BRW     CNX$SCAN_CSBS_FORK         ; Fork and resume scan
                       09DB
                  05   09DB   40$:     RSB
                       09DC
          F621'   31   09DC   50$:     BRW     CNX$SCAN_CSBS_RETRY        ; Wait and resume scan
                       09DF
           0569   30   09DF   60$:     BSBW    DESCRIBE_NODE              ; Tell one node about another
      54   64 A3  D0   09E2            MOVL    CSB$L_CLUB(R3),R4          ; Address of CLUB
           03 50  E9   09E6            BLBC    R0,70$                     ; Error sending message
           05 51  E8   09E9            BLBS    R1,80$                     ; Branch if node is known
  00 1C A4   11   E2   09EC   70$:     BBSS    #CLUB$V_UNLOCK, -          ; Request that all be unlocked
                       09F1                    CLUB$L_FLAGS(R4),80$
           50 55  D0   09F1   80$:     MOVL    R5,R0
   00000000'GF    16   09F4            JSB     G^EXE$DEANONPAGED
                       09FA   ;
```

```
                                09FA  ; terminate threads -- except for the last one
                                09FA  ;
                 5A A4    B7    09FA  130$:   DECW    CLUB$W_MSGCNT(R4)
                    01    19    09FD          BLSS    140$                    ; Branch when done
                    05          09FF          RSB                             ; Terminate thread
                                0A00
                  0831    30    0A00  140$:   BSBW    CNX$CHECK_UNLOCK        ; Handle unlock request
                                0A03  ;
                                0A03  ; Now give the new node the information it needs to fill in its
                                0A03  ; Cluster vector.  The occupied slots will be taken care of in
                                0A03  ; the normal course of events.  The available slots are given
                                0A03  ; the value of the last sequence number used in this sequence
                                0A03  ; of instructions.
                                0A03  ;
     53      00000000'GF  3C    0A03          MOVZWL  G^CLU$GW_MAXINDEX,R3    ; Number of vector slots
                    39    11    0A0A          BRB     240$
                                0A0C
                 F5F1'    30    0A0C  200$:   BSBW    CNX$CLUB_WAIT           ; Wait a second, saving only R3
     51      00000000'GF  D0    0A0F  210$:   MOVL    G^CLU$GL_CLUSVEC,R1     ; Address of cluster vector
                  6143    D5    0A16          TSTL    (R1)[R3]                ; Contents of first slot
                    2A    19    0A19          BLSS    240$                    ; Found CSB -- slot in use
                 F5E2'    30    0A1B          BSBW    CNX$CLUB_FORK           ; Release control
                  0813    30    0A1E          BSBW    CNX$CHECK_UNLOCK        ; Bail out if unlock has been requested
                 F5DC'    30    0A21          BSBW    CNX$ALLOC_CDRP_ONLY     ; Get a fork block
                 F5D9'    30    0A24          BSBW    CNX$RESOURCE_CHECK      ; Watch out for exhausted resources
                 E2 50    E9    0A27          BLBC    R0,200$                 ; Branch if CDRP allocated
                    53    DD    0A2A          PUSHL   R3                      ; Save scan context
             53    70 A4  D0    0A2C          MOVL    CLUB$L_CTX0(R4),R3      ; Address of joining node CSB
             50       06  9A    0A30          MOVZBL  #CLMCNX$K_FNC_VEC,R0    ; Cluster vector description
                  08DC    30    0A33          BSBW    INIT_STD_MSG            ; Standard message initialization
      4C A5      0F04'CF  9E    0A36          MOVAB   W^BLD_VEC_MSG, -        ; Vector message building routine
                                0A3C                  CDRP$L_MSGBLD(R5)
          48 A5       6E  D0    0A3C          MOVL    (SP),CDRP$L_VAL8(R5)    ; Store Cluster vector index
                 F5BD'    30    0A40          BSBW    CNX$SEND_FORGET         ; Queue message and forget it
                    08    BA    0A43          POPR    #^M<R3>                 ; Restore context for scanning
                 C7 53    F4    0A45  240$:   SOBGEQ  R3,210$                 ; Iterate over all slots
     54      00000000'GF  D0    0A48          MOVL    G^CLU$GL_CLUB,R4        ; Address of CLUB
                                0A4F  ;
                                0A4F  ; Get here when all nodes have been described to all others
                                0A4F  ; Now have told every node about every other node and each node's
                                0A4F  ; CSB contains a map of nodes that can be seen
                                0A4F  ;
                                0A4F  JOIN_PROPOSE:
 00EC C4    20    00    6E    00    2C    0A4F          MOVC5   #0,(SP),#0, -           ; Init map of nodes that are
                                0A57                  #CLUB$S_NODEMAP, -      ;    totally connected
                                0A57                  CLUB$B_NODEMAP(R4)
                 F5A6'    30    0A57          BSBW    CNX$SCAN_CSBS           ; Scan all CSB's
                 10 50    E9    0A5A          BLBC    R0,240$                 ; Branch when done
          0A 60 A3   11   E1    0A5D          BBC     #CSB$V_SELECTED, -      ; Branch if not selected
                                0A62                  CSB$L_STATUS(R3),210$
             50    4C A3  3C    0A62          MOVZWL  CSB$W_CSID_IDX(R3),R0   ; CSID index of CSB
       00 00EC C4    50   E2    0A66          BBSS    R0,CLUB$B_NODEMAP(R4), - ; Node is part of proposed cluster
                                0A6C                  210$
                    05          0A6C  210$:   RSB                             ; Return and resume scan
                                0A6D
                                0A6D  240$:
                 F590'    30    0A6D          BSBW    CNX$DIRVEC_ADJ          ; Adjust size of lock manager directory vect
```

```
                 03 50    E8  0A70             BLBS    R0,250$                    ; Branch on success
                  07D0    31  0A73             BRW     UNLOCK_ALL                 ; Can't adjust vector size, abort
                              0A76  250$:
                              0A76  ;
                              0A76  ; Propose cluster -- Send Phase 1 Messages
                              0A76  ;
           70 A4    09    9A  0A76             MOVZBL  #CLMCNX$K_FNC_JOIN, -      ; Facility specific message code
                              0A7A                     CLUB$L_CTX0(R4)
       78 A4    0BFA'CF   9E  0A7A             MOVAB   W^BLD_JOIN_MSG, -          ; Address of routine to build
                              0A80                     CLUB$L_CTXT(R4)            ;  status message
                  05DC    30  0A80             BSBW    SEND_PH1                   ; Send phase 1 messages
                              0A83  ;
                              0A83  ; All nodes have seen Phase 1 and ACKed it.
                              0A83  ; Now is the time to send Phase 2.
                              0A83  ;
                  0638    31  0A83             BRW     SEND_PH2
                              0A86  ;
                              0A86  ; At the completion of sending Phase 2 messages, this location
                              0A86  ; is executed as a thread.
                              0A86  ;
                              0A86  JOIN_FINISH:
                              0A86  FORM_JOIN_FINISH:
           52    10 A4    D0  0A86             MOVL    CLUB$L_LOCAL_CSB(R4),R2 ; Local node CSB
 008C C2   00EC C4    20  28  0A8A             MOVC3   #CSB$S_NODEMAP, -          ; Copy membership map from CLUB
                              0A92                     CLUB$B_NODEMAP(R4), -      ;   to local node's CSB
                              0A92                     CSB$B_NODEMAP(R2)
                 F56B'    30  0A92             BSBW    CNX$SCAN_CSBS              ; Iterate over all CSBs
                 2A 50    E9  0A95             BLBC    R0,20$                     ; Branch when done
        24 60 A3    11    E1  0A98             BBC     #CSB$V_SELECTED, -         ; Branch if not selected
                              0A9D                     CSB$L_STATUS(R3),10$
        1F 60 A3    18    E0  0A9D             BBS     #CSB$V_LOCAL, -            ; Branch if local node
                              0AA2                     CSB$L_STATUS(R3),10$
                    18    BB  0AA2             PUSHR   #^M<R3,R4>                 ; Save registers
 008C C3   00EC C4    20  28  0AA4             MOVC3   #CSB$S_NODEMAP, -          ; Copy membership map from CLUB
                              0AAC                     CLUB$B_NODEMAP(R4), -      ;   to selected node's CSB
                              0AAC                     CSB$B_NODEMAP(R3)
                    18    BA  0AAC             POPR    #^M<R3,R4>                 ; Restore registers
        0E 60 A3    00    E1  0AAE             BBC     #CSB$V_LONG_BREAK, -       ; Branch if no long break
                              0AB3                     CSB$L_STATUS(R3),10$
           52    10 A4    D0  0AB3             MOVL    CLUB$L_LOCAL_CSB(R4),R2 ; Get local CSB address
           50    4C A3    3C  0AB7             MOVZWL  CSB$W_CSID_IDX(R3),R0      ; Remote node CSID index
     00 008C C2    50     E5  0ABB             BBCC    R0,CSB$B_NODEMAP(R2),10$ ; Mark connection broken
                    05        0AC1  10$:       RSB
                              0AC2
                 F53B'    30  0AC2  20$:       BSBW    CNX$FIX_EPID               ; Add node ID to EPIDs
                  089B    30  0AC5             BSBW    UPDATE_QUORUM              ; Update quorum, votes, nodes
                 F535'    30  0AC8             BSBW    CNX$SCAN_CSBS              ; Iterate over CSBs
                 28 50    E9  0ACB             BLBC    R0,60$                     ; Branch when scan is done
        1D 60 A3    11    E1  0ACE             BBC     #CSB$V_SELECTED, -         ; Branch if CSB not selected
                              0AD3                     CSB$L_STATUS(R3),40$
        13 60 A3    01    E0  0AD3             BBS     #CSB$V_MEMBER, -           ; Skip if node is already a member
                              0AD8                     CSB$L_STATUS(R3),30$
                  08D1    30  0AD8             BSBW    ADD_NODE
        0B 60 A3    18    E1  0ADB             BBC     #CSB$V_LOCAL, -            ; Branch if not local node
                              0AE0                     CSB$L_STATUS(R3),30$       ;   and skip outputting message
           55    53       D0  0AE0             MOVL    R3,R5                      ; Address of CSB
        50    0000'CF     9E  0AE3             MOVAB   ADDNODE_MSG,R0             ; Node added to cluster message
```

```
                  F515'  30  0AE8  2346          BSBW    CNX$CONFIG_CHANGE        ; Make this state known
                  F512'  30  0AEB  2347 30$:     BSBW    CNX$MARK_UNLOCKED        ; Mark the CSB unlocked
                     05  11  0AEE  2348          BRB     50$
                             0AF0  2349
       20 60 A3      01  E0  0AF0  2350 40$:     BBS     #CSB$V_MEMBER, -         ; Branch on error case of non-selected
                             0AF5  2351                  CSB$L_STATUS(R3),100$    ;   member
                         05  0AF5  2352 50$:     RSB
                             0AF6  2353
                  F507'  30  0AF6  2354 60$:     BSBW    CNX$DIRVEC_FILL          ; Update contents of lock manager directory
                00AC C4  B6  0AF9  2355          INCW    CLUB$W_MEMSEQ(R4)        ; Advance membership state sequence number
                             0AFD  2356 ;
                             0AFD  2357 ; Add local node to cluster
                             0AFD  2358 ;
       0D 1C A4      00  E2  0AFD  2359          BBSS    #CLUB$V_CLUSTER, -       ; Mark node a cluster member and
                             0B02  2360                  CLUB$L_FLAGS(R4),90$     ;   branch if a member already
 1C A4     10000000 8F   C8  0B02  2361          BISL2   #CLUB$M_QUORUM, -        ; New nodes assume a quorum
                             0B0A  2362                  CLUB$L_FLAGS(R4)
       00 1C A4      12  E5  0B0A  2363          BBCC    #CLUB$V_NO_FORM, -       ; Clear formation inhibit bit
                             0B0F  2364                  CLUB$L_FLAGS(R4),90$
                  F4EE'  30  0B0F  2365 90$:     BSBW    CNX$MEMBERSHIP_CHANGE    ; Do processing for adding a node
                   0731  31  0B12  2366          BRW     UNLOCK_ALL               ; Make sure all nodes are unlocked
                             0B15  2367
                             0B15  2368 100$:    BUG_CHECK       CNXMGRERR,FATAL ; Consistency check -- unselected node is me
```

```
                         0B19          .SBTTL  ADJUST_QUORUM - Adjust Cluster Quorum and/or Quorum Disk Membership
                         0B19
                         0B19  ;++
                         0B19  ;
                         0B19  ; FUNCTIONAL DESCRIPTION:
                         0B19  ;
                         0B19  ;       This routine adjusts the cluster quorum and/or changes quorum disk membershi
                         0B19  ;       This is done in a synchronized manner using a 2 phase protocol.
                         0B19  ;
                         0B19  ; CALLING SEQUENCE:
                         0B19  ;
                         0B19  ;       JSB     ADJUST_QUORUM
                         0B19  ;       IPL is IPL$_SCS
                         0B19  ;
                         0B19  ; INPUT PARAMETERS:
                         0B19  ;
                         0B19  ;       NONE
                         0B19  ;
                         0B19  ; OUTPUT PARAMETERS:
                         0B19  ;
                         0B19  ;       NONE
                         0B19  ;
                         0B19  ; COMPLETION CODES:
                         0B19  ;
                         0B19  ;       NONE
                         0B19  ;
                         0B19  ; SIDE EFFECTS:
                         0B19  ;
                         0B19  ;       R0-R5 are destroyed
                         0B19  ;
                         0B19  ;--
                         0B19
                         0B19  ADJUST_QUORUM:
            0085   30    0B19          BSBW    INIT_TRANSITION              ; Initialization for becoming coordinator
     58 A4    04   90    0B1C          MOVB    #CLMCNX$K_XTN_QUORUM, -      ; RECONFIGure transaction
                         0B20                  CLUB$B_CUR_CODE(R4)
              55   D4    0B20          CLRL    R5                           ; No CSB address
   50    0000'CF   9E    0B22          MOVAB   QUORUM_MSG,R0                ; Initiating cluster reconfiguration message
           F4D6'   30    0B27          BSBW    CNX$CONFIG_CHANGE            ; Make this state known
           F4D3'   30    0B2A          BSBW    CNX$SCAN_CSBS
           0B 50   E9    0B2D          BLBC    R0,20$                       ; Branch when done
  05 60 A3    01   E1    0B30          BBC     #CSB$V_MEMBER, -             ; Ignore nodes that are not members of
                         0B35                  CSB$L_STATUS(R3),10$         ;   the local cluster
  00 60 A3    11   E3    0B35          BBCS    #CSB$V_SELECTED, -           ; Mark node selected
                         0B3A                  CSB$L_STATUS(R3),10$
              05         0B3A  10$:    RSB
                         0B3B
                         0B3B  20$:
                         0B3B  ;
                         0B3B  ; Selected nodes are all members of the cluster.
                         0B3B  ;
            05F9   30    0B3B          BSBW    LOCK_NODES                   ; Attempt to lock selected nodes
                         0B3E  ;
                         0B3E  ; Get here when all nodes in the cluster have been locked.
                         0B3E  ; Compute and send Phase 1 proposal messages.
                         0B3E  ;
           F4BF'   30    0B3E          BSBW    CNX$QUORUM_CALC              ; Calculate and store quorum related paramet
```

```
                       0B41                         ; R0 is computed number of votes
  25 1C A4    1B  E1   0B41        BBC     #CLUB$V_ADJ_QUORUM, -   ; Branch if no quorum adjustment requested
                       0B46                CLUB$L_FLAGS(R4),60$
00A4 C4  00A6 C4  B0   0B46        MOVW    CLUB$W_ADJ_QUORUM(R4), - ; Copy proposed value to new cell
                       0B4D                CLUB$W_NEWQUORUM(R4)
  50     00A6 C4  B1   0B4D        CMPW    CLUB$W_ADJ_QUORUM(R4),R0 ; Minimize value with number of votes
              05  1F   0B52        BLSSU   30$                     ; Branch if new value would
  00A4 C4     50  B0   0B54        MOVW    R0,CLUB$W_NEWQUORUM(R4) ; Use current number of votes
           50 02  C0   0B59 30$:   ADDL2   #2,R0                   ; Compute (votes+2)/2
           50 02  C6   0B5C        DIVL2   #2,R0
  50     00A4 C4  B1   0B5F        CMPW    CLUB$W_NEWQUORUM(R4),R0 ; New quorum >= (votes+2)/2?
              05  1E   0B64        BGEQU   40$                     ; Branch if yes
  00A4 C4     50  B0   0B66        MOVW    R0,CLUB$W_NEWQUORUM(R4)
                       0B6B 40$:
                       0B6B 60$:
                       0B6B ;
                       0B6B ; Propose cluster -- Send Phase 1 Messages
                       0B6B ;
     70 A4    0D  9A   0B6B        MOVZBL  #CLMCNX$K_FNC_QUORUM, - ; Message code
                       0B6F                CLUB$L_CTX0(R4)
  78 A4  0BFA'CF  9E   0B6F        MOVAB   W^BLD_QUORUM_MSG, -     ; Address of routine to build
                       0B75                CLUB$L_CTX1(R4)         ;  status message
            04E7  30   0B75        BSBW    SEND_PH1                ; Send phase 1 messages
                       0B78 ;
                       0B78 ; Phase 1 has been sent.  Nothing can stop us now short of our own
                       0B78 ; failure.  Therefore, it is time to clear the bit requesting the
                       0B78 ; quorum adjustment.  This node has done its best...
                       0B78 ;
  00 1C A4    1B  E5   0B78        BBCC    #CLUB$V_ADJ_QUORUM, -   ; Clear quorum adjustment flag
                       0B7D                CLUB$L_FLAGS(R4),100$
                       0B7D 100$:
                       0B7D ;
                       0B7D ; All nodes have seen Phase 1 and ACKed it.
                       0B7D ; Now is the time to send Phase 2.
                       0B7D ;
            053E  31   0B7D        BRW     SEND_PH2
                       0B80 ;
                       0B80 ; At the completion of sending Phase 2 messages, this location
                       0B80 ; is executed as a thread.
                       0B80 ;
                       0B80 QUORUM_FINISH:
            07E0  30   0B80        BSBW    UPDATE_QUORUM           ; Update quorum, votes, nodes
                       0B83 ;
                       0B83 ; Reduce quorum in CSB's to prevent cluster quorum from ratcheting right
                       0B83 ; back up with the next state transition
                       0B83 ;
           F47A'  30   0B83        BSBW    CNX$SCAN_CSBS           ; Iterate over CSBs
           15 50  E9   0B86        BLBC    R0,230$                 ; Branch when scan is done
  0F 60 A3    01  E1   0B89        BBC     #CSB$V_MEMBER, -        ; Branch if node is cluster member
                       0B8E                CSB$L_STATUS(R3), 220$
  20 A4    52 A3  B1   0B8E        CMPW    CSB$W_QUORUM(R3), -     ; Compare sysgen parameter with current
                       0B93                CLUB$W_QUORUM(R4)       ;   cluster quorum
              05  1B   0B93        BLEQU   210$                    ; Branch if less than cluster quorum
  52 A3    20 A4  B0   0B95        MOVW    CLUB$W_QUORUM(R4), -    ; Lower copy of sysgen parameter to match
                       0B9A                CSB$W_QUORUM(R3)        ;   new, reduced cluster quorum
           F463'  30   0B9A 210$:  BSBW    CNX$MARK_UNLOCKED       ; Mark node no longer locked
              05       0B9D 220$:  RSB                             ; Continue iteration
```

```
                 0B9E  2484
06A5     31      0B9E  2485 230$:   BRW     UNLOCK_ALL              ; Make sure everything is unlocked
```

```
                                   0BA1        .SBTTL  INIT_TRANSITION - Initialization for a transition
                                   0BA1
                                   0BA1  ;++
                                   0BA1  ;
                                   0BA1  ; FUNCTIONAL DESCRIPTION:
                                   0BA1  ;
                                   0BA1  ;       This routine is the common initialization code for all
                                   0BA1  ;       transitions.  It is executed by any node which is attempting
                                   0BA1  ;       to become the coordinator.
                                   0BA1  ;
                                   0BA1  ; CALLING SEQUENCE:
                                   0BA1  ;
                                   0BA1  ;       JSB     INIT_TRANSITION
                                   0BA1  ;       IPL is IPL$_SCS
                                   0BA1  ;
                                   0BA1  ; INPUT PARAMETERS:
                                   0BA1  ;
                                   0BA1  ;       NONE
                                   0BA1  ;
                                   0BA1  ; OUTPUT PARAMETERS:
                                   0BA1  ;
                                   0BA1  ;       R3:     Address of local CSB
                                   0BA1  ;       R4:     Address of CLUB
                                   0BA1  ;
                                   0BA1  ; COMPLETION CODES:
                                   0BA1  ;
                                   0BA1  ;       NONE
                                   0BA1  ;
                                   0BA1  ; SIDE EFFECTS:
                                   0BA1  ;
                                   0BA1  ;       R0-R2,R5 are destroyed
                                   0BA1  ;
                                   0BA1  ;--
                                   0BA1
                                   0BA1  INIT_TRANSITION:
      54  00000000'GF      D0      0BA1          MOVL    G^CLU$GL_CLUB,R4          ; Address of CLUB
   1C A4  20000000 8F      C8      0BA8          BISL2   #CLUB$M_TRANSITION, -     ; Mark transition started
                                   0BB0                  CLUB$L_FLAGS(R4)
              0084 C4      D4      0BB0          CLRL    CLUB$L_TQE(R4)            ; Abandon any timeout in progress
        50    00CC C4      9E      0BB4          MOVAB   CLUB$B_FORK_BLOCK(R4),R0 ; Address of transition fork block
     03 1C A0      00      E5      0BB9          BBCC    #CLUBFKB$V_FKB_BUSY, -    ; Branch if fork block is not
                                   0BBE                  CLUBFKB$L_STATUS(R0),10$ ;   busy and mark no busy
                50    60   0F      0BBE          REMQUE  CLUBFKB$B_FORK_BLOCK(R0),R0 ; Remove from queue
   1C A4  00020000 8F      CA      0BC1  10$:    BICL2   #CLUB$M_UNLOCK, -         ; Clear unlock request
                                   0BC9                  CLUB$L_FLAGS(R4)
                 68 A4     D6      0BC9  20$:    INCL    CLUB$L_MAX_XTN(R4)        ; Compute transaction ID
                    FB     13      0BCC          BEQL    20$                       ; Disallow transition 0
     48 A4       68 A4     D0      0BCE          MOVL    CLUB$L_MAX_XTN(R4), -
                                   0BD3                  CLUB$L_CUR_XTN(R4)
        53       10 A4     D0      0BD3          MOVL    CLUB$L_LOCAL_CSB(R4),R3 ; CSB for this node
     4C A4       4C A3     D0      0BD7          MOVL    CSB$L_CSID(R3), -         ; CSID of coordinator
                                   0BDC                  CLUB$L_CUR_COORD(R4)
        5C A4       53     D0      0BDC          MOVL    R3,CLUB$L_COORD(R4)       ; Mark this node the coordinator
        59 A4       20     90      0BE0          MOVB    #CLMCNX$K_LOCK, -         ; Set transaction phase
                                   0BE4                  CLUB$B_CUR_PHASE(R4)
   50 A4  00000000'GF      7D      0BE4          MOVQ    G^EXE$GQ_SYSTIME, -       ; Current time is timestamp for
                                   0BEC                  CLUB$Q_COR_TIME(R4)       ;   this transition
```

```
64 A4    66 A4   B0  0BEC  2544          MOVW    CLUB$W_FIRST_INDEX(R4), - ; Initialize CSID allocation context
                     0BF1  2545                  CLUB$W_NEXT_CSID(R4)
           F40C'  30  0BF1  2546          BSBW    CNX$INIT_CSBS             ; Initialize CSBs for transition
00 60 A3     11   E3  0BF4  2547          BBCS    #CSB$V_SELECTED, -        ; Mark local node selected
                     0BF9  2548                  CSB$L_STATUS(R3),30$
                  05  0BF9  2549 30$:     RSB
```

```
                     0BFA           .SBTTL  BLD_FORM_MSG - Build Message Proposing a New Cluster
                     0BFA           .SBTTL  BLD_RECONFIG_MSG - Build Message Proposing a Reconfigured Cluster
                     0BFA           .SBTTL  BLD_JOIN_MSG - Build Message Proposing a New Node in a Cluster
                     0BFA           .SBTTL  BLD_QUORUM_MSG - Build Message Proposing a Quorum/Quorum Disk Chanag
                     0BFA      ;++
                     0BFA      ;
                     0BFA      ; FUNCTIONAL DESCRIPTION:
                     0BFA      ;
                     0BFA      ;       This routine builds a message proposing the formation, reconfiguration, or
                     0BFA      ;       addition of a node to a cluster.
                     0BFA      ;       It uses the standard message building routine to build the message header.
                     0BFA      ;
                     0BFA      ; CALLING SEQUENCE:
                     0BFA      ;
                     0BFA      ;       JSB     BLD_FORM_MSG
                     0BFA      ;       JSB     BLD_RECONFIG_MSG
                     0BFA      ;       JSB     BLD_JOIN_MSG
                     0BFA      ;       JSB     BLD_QUORUM_MSG
                     0BFA      ;       IPL is IPL$_SCS
                     0BFA      ;
                     0BFA      ; INPUT PARAMETERS:
                     0BFA      ;
                     0BFA      ;       R2:     Address of message buffer
                     0BFA      ;       R3:     Address of CSB
                     0BFA      ;       R4:     Address of PDT
                     0BFA      ;       R5:     Address of CDRP
                     0BFA      ;               CDRP$L_VAL1(R5): Byte 0 contains facility code (negated for response
                     0BFA      ;                                Byte 1 contains facility specific function code
                     0BFA      ;               CDRP$L_VAL2(R5): Transition ID (from CLUB$L_CUR_XTN)
                     0BFA      ;               CDRP$L_VAL3(R5): Byte 0 contains transition phase (from CLUB$B_CUR_P
                     0BFA      ;                                Byte 1 contains transition code (from CLUB$B_XTN_CO
                     0BFA      ;                                Byte 2 contains success/failure flag
                     0BFA      ;                                Byte 3 contains reply code
                     0BFA      ;
                     0BFA      ; OUTPUT PARAMETERS:
                     0BFA      ;
                     0BFA      ;       NONE
                     0BFA      ;
                     0BFA      ; COMPLETION CODES:
                     0BFA      ;
                     0BFA      ;       NONE
                     0BFA      ;
                     0BFA      ; SIDE EFFECTS:
                     0BFA      ;
                     0BFA      ;       R0 and R1 are destroyed
                     0BFA      ;
                     0BFA      ;--
                     0BFA
                     0BFA      BLD_FORM_MSG:
                     0BFA      BLD_RECONFIG_MSG:
                     0BFA      BLD_JOIN_MSG:
                     0BFA      BLD_QUORUM_MSG:
         073C   30   0BFA           BSBW    BLD_STD_MSG             ; Fill in standard fields
      007C 8F   BB   0BFD           PUSHR   #^M<R2,R3,R4,R5,R6>     ; Save registers
   56   64 A3   D0   0C01           MOVL    CSB$L_CLUB(R3),R6       ; Address of CLUB
14 A2   64 A6   B0   0C05           MOVW    CLUB$Q_NEXT_CSID(R6), - ; Save CSID assignment context
                     0C0A                   CLMPRO$W_NEXT_CSID(R2)
```

```
         16 A2    00A4 C6   B0  0C0A  2608          MOVW    CLUB$W_NEWQUORUM(R6), - ; Cluster quorum
                                0C10  2609                  CLMPRO$W_QUORUM(R2)
         18 A2    00AC C6   B0  0C10  2610          MOVW    CLUB$W_MEMSEQ(R6), -    ; Cluster membership state sequence
                                0C16  2611                  CLMPRO$W_MEMSEQ(R2)     ;   number
         1C A2    00A8 C6   D0  0C16  2612          MOVL    CLUB$L_FMERIT(R6), -    ; Proposed cluster figure of merit
                                0C1C  2613                  CLMPRO$L_FMERIT(R2)
           20 A2    2C A6   7D  0C1C  2614          MOVQ    CLUB$Q_FTIME(R6), -     ; Cluster founding time
                                0C21  2615                  CLMPRO$Q_FTIME(R2)
      28 A2    00000000'GF  7D  0C21  2616          MOVQ    G^EXE$GQ_SYSTIME, -     ; Store current system time in
                                0C29  2617                  CLMPRO$Q_CURTIME(R2)    ;   the message
           1A A2    46 A6   B0  0C29  2618          MOVW    CLUB$W_NEWQDVOTES(R6), -      ; Quorum disk votes
                                0C2E  2619                  CLMPRO$W_QDVOTES(R2)
                    30 A2   94  0C2E  2620          CLRB    CLMPRO$B_FLAGS(R2)      ; Zero flags byte
           04 1C A6    1A   E1  0C31  2621          BBC     #CLUB$V_QF_NEWVOTE, -   ; Branch if quorum disk not part of
                                0C36  2622                  CLUB$L_FLAGS(R6),10$
                 30 A2  01  88  0C36  2623          BISB    #CLMPRO$M_QF_VOTE, -    ; Mark quorum disk present
                                0C3A  2624                  CLMPRO$B_FLAGS(R2)
      31 A2    26 A6    06  28  0C3A  2625 10$:     MOVC3   #CLMPRO$S_FSYSID, -     ; Cluster founding system
                                0C40  2626                  CLUB$B_FSYSID(R6), -
                                0C40  2627                  CLMPRO$B_FSYSID(R2)
                    52  6E  D0  0C40  2628          MOVL    (SP),R2                 ; Restore message buffer address
   37 A2    00EC C6     20  28  0C43  2629          MOVC3   #CLMPRO$S_NODEMAP, -    ; Fill map of proposed systems
                                0C4A  2630                  CLUB$B_NODEMAP(R6), -
                                0C4A  2631                  CLMPRO$B_NODEMAP(R2)
                  007C 8F   BA  0C4A  2632          POPR    #^M<R2,R3,R4,R5,R6>     ; Restore registers
                            05  0C4E  2633          RSB
```

```
                              0C4F          .SBTTL  CNX$RCVD_FORM - Cluster formation proposal received
                              0C4F
                              0C4F    ;++
                              0C4F    ;
                              0C4F    ; FUNCTIONAL DESCRIPTION:
                              0C4F    ;
                              0C4F    ;       This routine is called cluster formation proposal message is received.
                              0C4F    ;       A response must be sent which ACKs or NAKs the request.  The
                              0C4F    ;       request is NAKd if this node does not have a connection to all proposed
                              0C4F    ;       member nodes.
                              0C4F    ;
                              0C4F    ; CALLING SEQUENCE:
                              0C4F    ;
                              0C4F    ;       JSB     CNX$RCVD_FORM
                              0C4F    ;       IPL is IPL$_SCS
                              0C4F    ;
                              0C4F    ; INPUT PARAMETERS:
                              0C4F    ;
                              0C4F    ;       R2:     Message address
                              0C4F    ;       R3:     CSB of sending system
                              0C4F    ;       R4:     PDT address
                              0C4F    ;       R5:     CDRP address (uninitialized)
                              0C4F    ;
                              0C4F    ; OUTPUT PARAMETERS:
                              0C4F    ;
                              0C4F    ;       NONE
                              0C4F    ;
                              0C4F    ; COMPLETION CODES:
                              0C4F    ;
                              0C4F    ;       NONE
                              0C4F    ;
                              0C4F    ; SIDE EFFECTS:
                              0C4F    ;
                              0C4F    ;       R0-R5 may be destroyed.
                              0C4F    ;--
                              0C4F
                              0C4F    CNX$RCVD_FORM::
                      2C  BB  0C4F          PUSHR   #^M<R2,R3,R5>           ; Save needed registers
                   F3AC'  30  0C51          BSBW    CNX$INIT_CDRP           ; Initialize the CDRP for the response
                   52  6E  D0  0C54          MOVL    (SP),R2                 ; Restore message address
                    06EF  30  0C57          BSBW    MSG_CHECK               ; Validate message
        64 A4      14 A2  B0  0C5A          MOVW    CLMPRO$W_NEXT_CSID(R2), - ; Save CSID assignment context
                              0C5F                  CLUB$W_NEXT_CSID(R4)
      00A4 C4      16 A2  B0  0C5F          MOVW    CLMPRO$W_QUORUM(R2), -  ; Cluster quorum
                              0C65                  CLUB$W_NEWQUORUM(R4)
      00AC C4      18 A2  B0  0C65          MOVW    CLMPRO$W_MEMSEQ(R2), -  ; Cluster membership state sequence
                              0C6B                  CLUB$W_MEMSEQ(R4)       ;  number
      00A8 C4      1C A2  D0  0C6B          MOVL    CLMPRO$L_FMERIT(R2), -  ; Proposed cluster figure of merit
                              0C71                  CLUB$L_FMERIT(R4)
        2C A4      20 A2  7D  0C71          MOVQ    CLMPRO$Q_FTIME(R2), -   ; Cluster foundation time
                              0C76                  CLUB$Q_FTIME(R4)
      0094 C4      28 A2  7D  0C76          MOVQ    CLMPRO$Q_CURTIME(R2), - ; Save time from message
                              0C7C                  CLUB$Q_NEWTIME(R4)
        46 A4      1A A2  B0  0C7C          MOVW    CLMPRO$W_QDVOTES(R2), - ; Quorum disk votes
                              0C81                  CLUB$W_NEWQDVOTES(R4)
  1C A4    04000000 8F  CA  0C81          BICL    #CLUB$M_QF_NEWVOTE, -   ; Assume no quorum disk membership
                              0C89                  CLUB$L_FLAGS(R4)
```

```
      0C 30 A2    00   E1  0C89       BBC    #CLMPRO$V_QF_VOTE, -        ; Branch if no quorum disk membership
                           0C8E              CLMPRO$B_FLAGS(R2),10$
                  51   D4  0C8E       CLRL   R1                          ; Assume failure
      4A 1C A4    01   E1  0C90       BBC    #CLUB$V_QF_ACTIVE, -        ; Branch if disk not locally active
                           0C95              CLUB$L_FLAGS(R4),50$        ;   and reject proposal
      00 1C A4    1A   E2  0C95       BBSS   #CLUB$V_QF_NEWVOTE, -       ; Set quorum disk membership
                           0C9A              CLUB$L_FLAGS(R4),10$
                  1C   BB  0C9A  10$: PUSHR  #^M<R2,R3,R4>               ; Save registers
   26 A4    31 A2 06   28  0C9C       MOVC3  #CLUB$S_FSYSID, -           ; Cluster founding system ID
                           0CA2              CLMPRO$B_FSYSID(R2), -
                           0CA2              CLUB$B_FSYSID(R4)
                  1C   BA  0CA2       POPR   #^M<R2,R3,R4>               ; Restore registers
                  1C   BB  0CA4       PUSHR  #^M<R2,R3,R4>               ; Save registers
 00EC C4    37 A2 20   28  0CA6       MOVC3  #CLUB$S_NODEMAP, -          ; Save map of cluster nodes
                           0CAD              CLMPRO$B_NODEMAP(R2), -     ;   in CLUB
                           0CAD              CLUB$B_NODEMAP(R4)
                  1C   BA  0CAD       POPR   #^M<R2,R3,R4>               ; Restore registers
 009C C4    00000000'GF 7D 0CAF       MOVQ   G^EXE$GQ_SYSTIME, -         ; Save current system time
                           0CB8              CLUB$Q_NEWTIME_REF(R4)      ;   as a reference
            51    01   D0  0CB8       MOVL   #1,R1                       ; Anticipate success
                F342'  30  0CBB       BSBW   CNX$SCAN_CSBS               ; Iterate over all CSBs
               1E 50   E9  0CBE       BLBC   R0,50$                      ; Branch when done
      18 60 A3    11   E1  0CC1       BBC    #CSB$V_SELECTED, -          ; Branch if node not selected
                           0CC6              CSB$L_STATUS(R3),40$
         50    4C A3   3C  0CC6       MOVZWL CSB$W_CSID_IDX(R3),R0       ; CSID index
      0A 37 A2    50   E5  0CCA       BBCC   R0, -                       ; Branch if node is not in proposed
                           0CCF              CLMPRO$B_NODEMAP(R2), -     ;   cluster
                           0CCF              30$
      0A 60 A3    00   E1  0CCF       BBC    #CSB$V_LONG_BREAK, -        ; Branch if no long break seen
                           0CD4              CSB$L_STATUS(R3),40$
                  51   D4  0CD4       CLRL   R1
                F327'  31  0CD6       BRW    CNX$SCAN_CSBS_EXIT          ; Exit from CSB scanning loop
                           0CD9
      00 60 A3    11   E4  0CD9  30$: BBSC   #CSB$V_SELECTED, -          ; Clear selected bit
                           0CDE              CSB$L_STATUS(R3),40$
                       05  0CDE  40$: RSB
                           0CDF
                  51   DD  0CDF  50$: PUSHL  R1                          ; Save status
      37 A2    20 00   3B  0CE1       SKPC   #0,#CLMPRO$S_NODEMAP, -     ; Look for any missing nodes
                           0CE6              CLMPRO$B_NODEMAP(R2)
                  07   13  0CE6       BEQL   60$                         ; All nodes accounted for
      02 1C A4    11   E1  0CE8       BBC    #CLUB$V_UNLOCK, -           ; Branch if no unlock request
                           0CED              CLUB$L_FLAGS(R4), 60$       ;   pending
                  6E   D4  0CED       CLRL   (SP)                        ; NAK request
               09 6E   E9  0CEF  60$: BLBC   (SP),70$                    ; Branch on NAK pending
                F30B'  30  0CF2       BSBW   CNX$DIRVEC_ADJ              ; Adjust size of lock manager directory vect
               03 50   E8  0CF5       BLBS   R0,70$                      ; Branch on success
            6E    50   D0  0CF8       MOVL   R0,(SP)                     ; Make sure proposal is NAKed
                  2D   BA  0CFB  70$: POPR   #^M<R0,R2,R3,R5>            ; Restore needed registers
                  50   DD  0CFD       PUSHL  R0                          ; Save status
            50    07   9A  0CFF       MOVZBL #CLMCNX$K_FNC_FORM,R0       ; Facility specific function code
                0604   30  0D02       BSBW   CNX$INIT_STD_RESP           ; Init CDRP for standard response
         36 A5    8E   F6  0D05       CVTLB  (SP)+,CDRP$L_VAL3+2(R5)     ; Store success/failure flag
                F2F4'  31  0D09       BRW    CNX$RESP_FORGET             ; Queue response message
```

```
                          0D0C        .SBTTL  CNX$RCVD_RECONFIG - Cluster Reconfiguration proposal received
                          0D0C
                          0D0C  ;++
                          0D0C  ;
                          0D0C  ; FUNCTIONAL DESCRIPTION:
                          0D0C  ;
                          0D0C  ;       This routine is called when a cluster reconfiguration proposal
                          0D0C  ;       message is received.
                          0D0C  ;       A response must be sent which ACKs or NAKs the request.  The
                          0D0C  ;       request is NAKd if this node does not have a connection to exactly
                          0D0C  ;       the set of member nodes of the proposed reconfigured cluster.
                          0D0C  ;
                          0D0C  ; CALLING SEQUENCE:
                          0D0C  ;
                          0D0C  ;       JSB     CNX$RCVD_RECONFIG
                          0D0C  ;       IPL is IPL$_SCS
                          0D0C  ;
                          0D0C  ; INPUT PARAMETERS:
                          0D0C  ;
                          0D0C  ;       R2:     Message address
                          0D0C  ;       R3:     CSB of sending system
                          0D0C  ;       R4:     PDT address
                          0D0C  ;       R5:     CDRP address (uninitialized)
                          0D0C  ;
                          0D0C  ; OUTPUT PARAMETERS:
                          0D0C  ;
                          0D0C  ;       NONE
                          0D0C  ;
                          0D0C  ; COMPLETION CODES:
                          0D0C  ;
                          0D0C  ;       NONE
                          0D0C  ;
                          0D0C  ; SIDE EFFECTS:
                          0D0C  ;
                          0D0C  ;       R0-R5 may be destroyed.
                          0D0C  ;--
                          0D0C
                          0D0C  CNX$RCVD_RECONFIG::
                 2C   BB  0D0C          PUSHR   #^M<R2,R3,R5>           ; Save needed registers
              F2EF'   30  0D0E          BSBW    CNX$INIT_CDRP           ; Initialize the CDRP for the response
            52   6E   7D  0D11          MOVQ    (SP),R2                 ; Restore message address
               0632   30  0D14          BSBW    MSG_CHECK               ; Validate message
1C A4   00800000 8F   C8  0D17          BISL2   #CLUB$M_LOST_CNX, -     ; Mark connection to member lost
                          0D1F                  CLUB$L_FLAGS(R4)
    00000000'GF  01   8E  0D1F          MNEGB   #1,G^LCK$GB_STALLREQS   ; Stall lock requests
   00A8 C4    1C A2   D0  0D26          MOVL    CLMPRO$L_FMERIT(R2), -  ; Proposed cluster figure of merit
                          0D2C                  CLUB$L_FMERIT(R4)
 00EC C4   37 A2  20  28  0D2C          MOVC3   #CLMPRO$S_NODEMAP, -    ; Save map of cluster nodes
                          0D33                  CLMPRO$B_NODEMAP(R2), - ;   in CLUB
                          0D33                  CLUB$B_NODEMAP(R4)
                          0D33
            51   01   D0  0D33          MOVL    #1,R1                   ; Status indicator
            52   6E   D0  0D36          MOVL    (SP),R2                 ; Message buffer address
              F2C4'   30  0D39          BSBW    CNX$SCAN_CSBS           ; Iterate over all CSBs
            1E   50   E9  0D3C          BLBC    R0,20$                  ; Branch when iteration is complete
      18 60 A3   01   E1  0D3F          BBC     #CSB$V_MEMBER, -        ; Branch if node is not a cluster member
                          0D44                  CSB$L_STATUS(R3),10$
```

```
60 A3    00020000 8F   C8  0D44         BISL2   #CSB$M_SELECTED, -            ; Select every member
                           0D4C                 CSB$L_STATUS(R3)
        50    4C A3    3C  0D4C         MOVZWL  CSB$W_CSID_IDX(R3),R0         ; System index
    07 37 A2      50   E5  0D50         BBCC    R0, -                         ; Branch if member not in proposed cluster
                           0D55                 CLMPRO$B_NODEMAP(R2),10$
    02 60 A3      00   E1  0D55         BBC     #CSB$V_LONG_BREAK, -          ; Branch if no long break seen
                           0D5A                 CSB$L_STATUS(R3),10$
                  51   D4  0D5A         CLRL    R1                            ; Force failure of proposal
                  05   05  0D5C  10$:   RSB
                           0D5D
               26 51   E9  0D5D  20$:   BLBC    R1,25$                        ; Branch on previous decision to reject
  00A4 C4      16 A2   B0  0D60         MOVW    CLMPRO$W_QUORUM(R2), -        ; Store quorum
                           0D66                 CLUB$W_NEWQUORUM(R4)
    46 A4      1A A2   B0  0D66         MOVW    CLMPRO$W_QDVOTES(R2), -       ; Store quorum disk votes
                           0D6B                 CLUB$W_NEWQDVOTES(R4)
1C A4    04000000 8F   CA  0D6B         BICL    #CLUB$M_QF_NEWVOTE, -         ; Assume no quorum disk membership
                           0D73                 CLUB$L_FLAGS(R4)
    0A 30 A2      00   E1  0D73         BBC     #CLMPRO$V_QF_VOTE, -          ; Branch if no quorum disk membership
                           0D78                 CLMPRO$B_FLAGS(R2),23$
    07 1C A4      01   E1  0D78         BBC     #CLUB$V_QF_ACTIVE, -          ; Branch if disk not locally active
                           0D7D                 CLUB$L_FLAGS(R4),24$          ;   and reject proposal
    00 1C A4      1A   E2  0D7D         BBSS    #CLUB$V_QF_NEWVOTE, -         ; Set quorum disk membership
                           0D82                 CLUB$L_FLAGS(R4),23$
                  02   11  0D82  23$:   BRB     25$
                           0D84
                  51   D4  0D84  24$:   CLRL    R1                            ; Reject proposal
                  51   DD  0D86  25$:   PUSHL   R1                            ; Save status return
 37 A2     20     00   3B  0D88         SKPC    #0,#CLMPRO$S_NODEMAP, -       ; Look for any missing nodes
                           0D8D                 CLMPRO$B_NODEMAP(R2)
                  04   13  0D8D         BEQL    30$                           ; All nodes accounted for
                           0D8F         BUG_CHECK       CNXMGRERR,FATAL       ; Some non-member node proposed as member
                           0D93
               31 6E   E9  0D93  30$:   BLBC    (SP),70$                      ; Reject proposal as inconsistent with local
    2A 1C A4      11   E0  0D96         BBS     #CLUB$V_UNLOCK, -             ; Branch if unlock request
                           0D9B                 CLUB$L_FLAGS(R4),60$          ;   is pending
               F262'   30  0D9B         BSBW    CNX$OPT                       ; Try to compute better subcluster
               03 50   E9  0D9E         BLBC    R0,40$                        ; Branch on computation failure
               21 51   E8  0DA1         BLBS    R1,60$                        ; Branch if a better cluster was computed
               F259'   30  0DA4  40$:   BSBW    CNX$DIRVEC_ADJ                ; Adjust size of lock manager directory vect
               1B 50   E9  0DA7         BLBC    R0,60$                        ; Branch on failure
               F253'   30  0DAA         BSBW    CNX$SCAN_CSBS                 ; Iterate over all CSBs
               17 50   E9  0DAD         BLBC    R0,70$                        ; Branch when iteration done
    0F 60 A3      01   E1  0DB0         BBC     #CSB$V_MEMBER, -              ; Branch if system is not a member
                           0DB5                 CSB$L_STATUS(R3),50$
        50    4C A3    3C  0DB5         MOVZWL  CSB$W_CSID_IDX(R3),R0         ; System ID index
 05 00EC C4       50   E0  0DB9         BBS     R0,CLUB$B_NODEMAP(R4),50$ ; Branch if system is in proposed cluster
    00 60 A3      11   E5  0DBF         BBCC    #CSB$V_SELECTED, -            ; Clear selected bit
                           0DC4                 CSB$L_STATUS(R3),50$
                  05   05  0DC4  50$:   RSB                                   ; End of iteration step
                           0DC5
                  6E   D4  0DC5  60$:   CLRL    (SP)                          ; Failure -- reject proposal
                  2D   BA  0DC7  70$:   POPR    #^M<R0,R2,R3,R5>              ; Restore status and needed registers
                  50   DD  0DC9         PUSHL   R0                            ; Save status
           50     08   9A  0DCB         MOVZBL  #CLMCNX$K_FNC_RECONFIG,R0           ; Facility specific function code
                0538   30  0DCE         BSBW    CNX$INIT_STD_RESP             ; Init CDRP for standard response
        36 A5     8E   F6  0DD1         CVTLB   (SP)+,CDRP$L_VAL3+2(R5)       ; Store success/failure flag
               F228'   31  0DD5         BRW     CNX$RESP_FORGET               ; Queue response message
```

```
                        0DD8         .SBTTL  CNX$RCVD_JOIN - Cluster node addition proposal received
                        0DD8
                        0DD8   ;++
                        0DD8   ;
                        0DD8   ; FUNCTIONAL DESCRIPTION:
                        0DD8   ;
                        0DD8   ;       This routine is called cluster node aaddition proposal message is received.
                        0DD8   ;       A response must be sent which ACKs or NAKs the request.  The
                        0DD8   ;       request is NAKd if this node does not have a connection to all proposed
                        0DD8   ;       member nodes.
                        0DD8   ;
                        0DD8   ; CALLING SEQUENCE:
                        0DD8   ;
                        0DD8   ;       JSB     CNX$RCVD_JOIN
                        0DD8   ;       IPL is IPL$_SCS
                        0DD8   ;
                        0DD8   ; INPUT PARAMETERS:
                        0DD8   ;
                        0DD8   ;       R2:     Message address
                        0DD8   ;       R3:     CSB of sending system
                        0DD8   ;       R4:     PDT address
                        0DD8   ;       R5:     CDRP address (uninitialized)
                        0DD8   ;
                        0DD8   ; OUTPUT PARAMETERS:
                        0DD8   ;
                        0DD8   ;       NONE
                        0DD8   ;
                        0DD8   ; COMPLETION CODES:
                        0DD8   ;
                        0DD8   ;       NONE
                        0DD8   ;
                        0DD8   ; SIDE EFFECTS:
                        0DD8   ;
                        0DD8   ;       R0-R5 may be destroyed.
                        0DD8   ;--
                        0DD8
                        0DD8   CNX$RCVD_JOIN::
             00    DD   0DD8           PUSHL   #0                          ; Anticipate NAK
             2C    BB   0DDA           PUSHR   #^M<R2,R3,R5>               ; Save needed registers
          F221'    30   0DDC           BSBW    CNX$INIT_CDRP               ; Initialize the CDRP for the response
          52  6E   D0   0DDF           MOVL    (SP),R2                     ; Restore message address
           0564    30   0DE2           BSBW    MSG_CHECK                   ; Validate message
   03 1C A4  17    E1   0DE5           BBC     #CLUB$V_LOST_CNX, -         ; Branch if node addition not inhibited
                        0DEA                   CLUB$L_FLAGS(R4),10$
           00A0    31   0DEA           BRW     70$                         ; Helper branch when node addition inhibited
   64 A4  14 A2    B0   0DED   10$:    MOVW    CLMPRO$W_NEXT_CSID(R2), -   ; Save CSID assignment context
                        0DF2                   CLUB$W_NEXT_CSID(R4)
 00AC C4  18 A2    B0   0DF2           MOVW    CLMPRO$W_MEMSEQ(R2), -      ; Cluster membership state sequence
                        0DF8                   CLUB$W_MEMSEQ(R4)           ;   number
   1A 1C A4  00    E0   0DF8           BBS     #CLUB$V_CLUSTER, -          ; Branch if node is already a cluster member
                        0DFD                   CLUB$L_FLAGS(R4),30$
                        0DFD   ;
                        0DFD   ; Receiving node is not cluster member
                        0DFD   ;
   2C A4  20 A2    7D   0DFD           MOVQ    CLMPRO$Q_FTIME(R2), -       ; Cluster foundation time
                        0E02                   CLUB$Q_FTIME(R4)
 0094 C4  28 A2    7D   0E02           MOVQ    CLMPRO$Q_CURTIME(R2), -     ; Save time from message
```

```
                                0E08             CLUB$Q_NEWTIME(R4)
      009C C4   00000000'GF  7D 0E08    MOVQ     G^EXE$GQ_SYSTIME, -      ; Save current system time
                                0E11             CLUB$Q_NEWTIME_REF(R4)   ;   as a reference
   26 A4    31 A2    06      28 0E11    MOVC3    #CLUB$S_FSYSID, -        ; Cluster founding system ID
                                0E17             CLMPRO$B_FSYSID(R2), -
                                0E17             CLUB$B_FSYSID(R4)
                    52    6E 7D 0E17 30$: MOVQ   (SP),R2                  ; Restore registers
               54    64 A3   D0 0E1A    MOVL     CSB$L_CLUB(R3),R4
          46 A4    1A A2     B0 0E1E    MOVW     CLMPRO$W_QDVOTES(R2), -  ; Quorum disk votes
                                0E23             CLUB$W_NEWQDVOTES(R4)
        00A4 C4    16 A2     B0 0E23    MOVW     CLMPRO$W_QUORUM(R2), -   ; Cluster quorum
                                0E29             CLUB$W_NEWQUORUM(R4)
      1C A4   04000000 8F    CA 0E29    BICL     #CLUB$M_QF_NEWVOTE, -    ; Assume no quorum disk membership
                                0E31             CLUB$L_FLAGS(R4)
          0C 30 A2    00     E1 0E31    BBC      #CLMPRO$V_QF_VOTE, -     ; Branch if no quorum disk membership
                                0E36             CLMPRO$B_FLAGS(R2),35$
                         51  D4 0E36    CLRL     R1                       ; Assume failure
          42 1C A4    01     E1 0E38    BBC      #CLUB$V_QF_ACTIVE, -     ; Branch if disk not locally active
                                0E3D             CLUB$L_FLAGS(R4),60$     ;   and reject proposal
          00 1C A4    1A     E2 0E3D    BBSS     #CLUB$V_QF_NEWVOTE, -    ; Set quorum disk membership
                                0E42             CLUB$L_FLAGS(R4),35$
      00EC C4    37 A2    20 28 0E42 35$: MOVC3  #CLUB$S_NODEMAP, -       ; Save map of cluster nodes
                                0E49             CLMPRO$B_NODEMAP(R2), -  ;   in CLUB
                                0E49             CLUB$B_NODEMAP(R4)
                    52    6E D0 0E49    MOVL     (SP),R2                  ; Restore message address
                    51    01 D0 0E4C    MOVL     #1,R1                    ; Anticipate success
                      F1AE'  30 0E4F    BSBW     CNX$SCAN_CSBS            ; Iterate over all CSBs
                      2A 50  E9 0E52    BLBC     R0,60$                   ; Branch when done
      60 A3   00020002 8F    D3 0E55    BITL     #CSB$M_SELECTED ! -      ; Check for SELECTED or
                                0E5D              CSB$M_MEMBER, -         ;   MEMBER
                                0E5D             CSB$L_STATUS(R3)
                         1F  13 0E5D    BEQL     50$                      ; Branch and ignore if neither
      60 A3   00020000 8F    C8 0E5F    BISL2    #CSB$M_SELECTED, -       ; Mark the node selected
                                0E67             CSB$L_STATUS(R3)
               50    4C A3   3C 0E67    MOVZWL   CSB$W_CSID_IDX(R3),R0    ; CSID index
          0A 37 A2    50     E5 0E6B    BBCC     R0, -                    ; Branch if node is not in proposed
                                0E70             CLMPRO$B_NODEMAP(R2), -  ;   cluster
                                0E70             40$
          09 60 A3    00     E1 0E70    BBC      #CSB$V_LONG_BREAK, -     ; Branch if no long break seen
                                0E75             CSB$L_STATUS(R3),50$
                         51  D4 0E75    CLRL     R1                       ; Flag failure
                      F186'  31 0E77    BRW      CNX$SCAN_CSBS_EXIT       ; Exit from CSB scanning loop
                                0E7A
                                0E7A 40$: BUG_CHECK       CNXMGRERR,FATAL ; Consistency check
                                0E7E
                             05 0E7E 50$: RSB
                                0E7F
              0C AE    51    D0 0E7F 60$: MOVL   R1,12(SP)                ; Save status
                      07 51  E9 0E83    BLBC     R1,70$                   ; Branch if NAK required
         37 A2    20         00 0E86    SKPC     #0,#CLMPRO$S_NODEMAP, -  ; Look for any missing nodes
                                0E8B             CLMPRO$B_NODEMAP(R2)
                         ED  12 0E8B    BNEQ     40$                      ; Some node(s) is not accounted for
                   0A 0C AE  E9 0E8D 70$: BLBC   12(SP),80$               ; Branch on NAK pending
                      F16C'  30 0E91    BSBW     CNX$DIRVEC_ADJ           ; Adjust size of lock manager directory vect
                      04 50  E8 0E94    BLBS     R0,80$                   ; Branch on success
              0C AE    50    D0 0E97    MOVL     R0,12(SP)                ; Make sure proposal is NAKed
                         2C  BA 0E9B 80$: POPR   #^M<R2,R3,R5>            ; Restore needed registers
```

```
      50    09   9A  0E9D  2976          MOVZBL  #CLMCNX$K_FNC_JOIN,R0    ; Facility specific function code
          0466   30  0EA0  2977          BSBW    CNX$INIT_STD_RESP        ; Init CDRP for standard response
   36 A5    8E   F6  0EA3  2978          CVTLB   (SP)+,CDRP$L_VAL3+2(R5)  ; Store success/failure flag
          F156'  31  0EA7  2979          BRW     CNX$RESP_FORGET          ; Queue response message
```

```
                        0EAA          .SBTTL  CNX$RCVD_QUORUM  - Quorum Update Message Received
                        0EAA
                        0EAA  ;++
                        0EAA  ;
                        0EAA  ; FUNCTIONAL DESCRIPTION:
                        0EAA  ;
                        0EAA  ;       This routine is called when a quorum update request message is received.
                        0EAA  ;       This is a Phase 1 message.  The validity of the message is checked and
                        0EAA  ;       the new quorum value is saved in a temporary location.
                        0EAA  ;
                        0EAA  ; CALLING SEQUENCE:
                        0EAA  ;
                        0EAA  ;       JSB     CNX$RCVD_QUORUM
                        0EAA  ;       IPL is IPL$_SCS
                        0EAA  ;
                        0EAA  ; INPUT PARAMETERS:
                        0EAA  ;
                        0EAA  ;       R2:     Message address
                        0EAA  ;       R3:     CSB of sending system
                        0EAA  ;       R4:     PDT address
                        0EAA  ;       R5:     CDRP address (uninitialized)
                        0EAA  ;
                        0EAA  ; OUTPUT PARAMETERS:
                        0EAA  ;
                        0EAA  ;       NONE
                        0EAA  ;
                        0EAA  ; COMPLETION CODES:
                        0EAA  ;
                        0EAA  ;       NONE
                        0EAA  ;
                        0EAA  ; SIDE EFFECTS:
                        0EAA  ;
                        0EAA  ;       R0-R5 may be destroyed.
                        0EAA  ;--
                        0EAA
                        0EAA  CNX$RCVD_QUORUM::
             2C     BB  0EAA          PUSHR   #^M<R2,R3,R5>                ; Save needed registers
          F151'     30  0EAC          BSBW    CNX$INIT_CDRP                ; Initialize the CDRP for the response
          52  6E    7D  0EAF          MOVQ    (SP),R2                      ; Restore message address and CSB address
           0494     30  0EB2          BSBW    MSG_CHECK                    ; Validate message
          F148'     30  0EB5          BSBW    CNX$SCAN_CSBS                ; Iterate over all CSBs
          0B  50    E9  0EB8          BLBC    R0,20$                       ; Branch when done
 05 60 A3    01     E1  0EBB          BBC     #CSB$V_MEMBER, -             ; Branch if not a member node
                        0EC0                  CSB$L_STATUS(R3),10$
 00 60 A3    11     E2  0EC0          BBSS    #CSB$V_SELECTED, -           ; Select each menber
                        0EC5                  CSB$L_STATUS(R3),10$
             05         0EC5  10$:    RSB
                        0EC6
          52  6E    7D  0EC6  20$:    MOVQ    (SP),R2                      ; Restore message buffer address
 00A4 C4   16 A2    B0  0EC9          MOVW    CLMPRO$W_QUORUM(R2), -       ; Temporarily save new quorum value
                        0ECF                  CLUB$W_NEWQUORUM(R4)
   46 A4   1A A2    B0  0ECF          MOVW    CLMPRO$W_QDVOTES(R2), -      ; Quorum disk votes
                        0ED4                  CLUB$W_NEWQDVOTES(R4)
1C A4  04000000 8F  CA  0ED4          BICL    #CLUB$M_QF_NEWVOTE, -        ; Assume no quorum disk membership
                        0EDC                  CLUB$L_FLAGS(R4)
          51  01    D0  0EDC          MOVL    #1,R1                        ; Assume that we will accept proposal
 0F 30 A2    00     E1  0EDF          BBC     #CLMPRO$V_QF_VOTE, -         ; Branch if no quorum disk membership
```

```
                        0EE4  3038                          CLMPRO$B_FLAGS(R2),30$
              51    D4  0EE4  3039          CLRL    R1                          ; Assume failure
  08 1C A4    01    E1  0EE6  3040          BBC     #CLUB$V_QF_ACTIVE, -        ; Branch if disk not locally active
                        0EEB  3041                  CLUB$L_FLAGS(R4),30$        ;   and reject proposal
              51    01  D0  0EEB  3042      MOVL    #1,R1                       ; Accept proposal
  00 1C A4    1A    E2  0EEE  3043          BBSS    #CLUB$V_QF_NEWVOTE, -       ; Set quorum disk membership
                        0EF3  3044                  CLUB$L_FLAGS(R4),30$
              2C    BA  0EF3  3045 30$:     POPR    #^M<R2,R3,R5>               ; Restore needed registers
              51    DD  0EF5  3046          PUSHL   R1                          ; Save status
        50    0D    9A  0EF7  3047          MOVZBL  #CLMCNX$K_FNC_QUORUM,R0     ; Facility specific function code
            040C    30  0EFA  3048          BSBW    CNX$INIT_STD_RESP           ; Init CDRP for standard response
     36 A5    8E    F6  0EFD  3049          CVTLB   (SP)+,CDRP$L_VAL3+2(R5)     ; Store success flag
           F0FC'    31  0F01  3050          BRW     CNX$RESP_FORGET             ; Send response message and forget it
```

```
                              0F04          .SBTTL  BLD_VEC_MSG - Build Message Describing Vector Slot
                              0F04
                              0F04     ;++
                              0F04     ;
                              0F04     ; FUNCTIONAL DESCRIPTION:
                              0F04     ;
                              0F04     ;        This routine builds a message describing an empty cluster vector slot.
                              0F04     ;        This is needed in order to guarantee consistent, cluster-wide assignment
                              0F04     ;        of unique CSIDs.
                              0F04     ;
                              0F04     ; CALLING SEQUENCE:
                              0F04     ;
                              0F04     ;        JSB     BLD_VEC_MSG
                              0F04     ;        IPL is IPL$_SCS
                              0F04     ;
                              0F04     ; INPUT PARAMETERS:
                              0F04     ;
                              0F04     ;        R2:     Address of message buffer
                              0F04     ;        R3:     Address of CSB
                              0F04     ;        R4:     Address of PDT
                              0F04     ;        R5:     Address of CDRP
                              0F04     ;                CDRP$L_VAL1(R5): Byte 0 contains facility code (negated for response
                              0F04     ;                                 Byte 1 contains facility specific function code
                              0F04     ;                CDRP$L_VAL2(R5): Transition ID (from CLUB$L_CUR_XTN)
                              0F04     ;                CDRP$L_VAL3(R5): Byte 0 contains transition phase (from CLUB$B_CUR_P
                              0F04     ;                                 Byte 1 contains transition code (from CLUB$B_XTN_CO
                              0F04     ;                                 Byte 2 contains success/failure flag
                              0F04     ;                                 Byte 3 contains reply code
                              0F04     ;
                              0F04     ; OUTPUT PARAMETERS:
                              0F04     ;
                              0F04     ;        NONE
                              0F04     ;
                              0F04     ; COMPLETION CODES:
                              0F04     ;
                              0F04     ;        NONE
                              0F04     ;
                              0F04     ; SIDE EFFECTS:
                              0F04     ;
                              0F04     ;        R0 and R1 are destroyed
                              0F04     ;
                              0F04     ;--
                              0F04
                              0F04     BLD_VEC_MSG:
              0432     30     0F04          BSBW    BLD_STD_MSG             ; Fill in standard fields
       50    48 A5     D0     0F07          MOVL    CDRP$L_VAL8(R5),R0      ; Slot index
       14 A2    50     F7     0F0B          CVTLW   R0,CLMVEC$W_INDEX(R2)   ; Store slot index
 51  00000000'GF       D0     0F0F          MOVL    G^CLU$GL_CLUSVEC,R1     ; Address of cluster vector
       50     6140     D0     0F16          MOVL    (R1)[R0],R0             ; Get contents of slot
                07     19     0F1A          BLSS    10$                     ; Branch if slot in use
       16 A2    50     F7     0F1C          CVTLW   R0,CLMVEC$W_SEQUENCE(R2) ; Last sequence number used
                01     1D     0F20          BVS     10$                     ; Branch on overflow
                       05     0F22          RSB
                              0F23
                              0F23     10$:  BUG_CHECK       CNXMGRERR,FATAL ; Consistency check
```

```
                             0F27           .SBTTL  CNX$RCVD_VEC - Cluster vector slot information received
                             0F27
                             0F27      ;++
                             0F27      ;
                             0F27      ; FUNCTIONAL DESCRIPTION:
                             0F27      ;
                             0F27      ;        This routine is called when a cluster vector slot description
                             0F27      ;        is received.
                             0F27      ;        No response or immediate acknowledgement is sent.
                             0F27      ;
                             0F27      ; CALLING SEQUENCE:
                             0F27      ;
                             0F27      ;        JSB     CNX$RCVD_VEC
                             0F27      ;        IPL is IPL$_SCS
                             0F27      ;
                             0F27      ; INPUT PARAMETERS:
                             0F27      ;
                             0F27      ;        R2:     Message address
                             0F27      ;        R3:     CSB of sending system
                             0F27      ;        R4:     PDT address
                             0F27      ;
                             0F27      ; OUTPUT PARAMETERS:
                             0F27      ;
                             0F27      ;        NONE
                             0F27      ;
                             0F27      ; COMPLETION CODES:
                             0F27      ;
                             0F27      ;        NONE
                             0F27      ;
                             0F27      ; SIDE EFFECTS:
                             0F27      ;
                             0F27      ;        R0-R5 may be destroyed.
                             0F27      ;--
                             0F27
                             0F27      CNX$RCVD_VEC::
              041F   30      0F27           BSBW    MSG_CHECK                ; Validate message
50    00000000'GF    D0      0F2A           MOVL    G^CLU$GL_CLUSVEC,R0      ; Address of cluster vector
        51   14 A2   3C      0F31           MOVZWL  CLMVEC$W_INDEX(R2),R1    ; Slot index
00000000'GF      51  B1      0F35           CMPW    R1,G^CLU$GW_MAXINDEX     ; Valid index?
                 09  1E      0F3C           BGEQU   10$                      ; Invalid index
      6041   16 A2   3C      0F3E           MOVZWL  CLMVEC$W_SEQUENCE(R2), - ; Store last used sequence number
                             0F43                   (R0)[R1]
             F0BA'   30      0F43           BSBW    CNX$DEALL_MSG_BUF_CSB    ; Deallocate message buffer
                     05      0F46           RSB                              ; Return
                             0F47
                             0F47      10$:    BUG_CHECK       CNXMGRERR,FATAL ; Consistency check
```

```
                       0F4B          .SBTTL  DESCRIBE_NODE - Describe one node to another node
                       0F4B
                       0F4B  ;++
                       0F4B  ;
                       0F4B  ; FUNCTIONAL DESCRIPTION:
                       0F4B  ;
                       0F4B  ;       This routine tells one node about another node.  It returns
                       0F4B  ;       asynchronously with information indicating that the first
                       0F4B  ;       node can or cannot see the second node.
                       0F4B  ;
                       0F4B  ; CALLING SEQUENCE:
                       0F4B  ;
                       0F4B  ;       JSB     DESCRIBE_NODE
                       0F4B  ;       IPL is IPL$_SCS
                       0F4B  ;
                       0F4B  ; INPUT PARAMETERS:
                       0F4B  ;
                       0F4B  ;       R5 is address of a CDRP
                       0F4B  ;       R3 is CSB address of node to talk to
                       0F4B  ;       R2 is CSB address of node to describe
                       0F4B  ;
                       0F4B  ; OUTPUT PARAMETERS:
                       0F4B  ;
                       0F4B  ;       R5 is address of a CDRP
                       0F4B  ;       R3 is CSB address of node talked to
                       0F4B  ;       R2 is CSB address of node described
                       0F4B  ;
                       0F4B  ; COMPLETION CODES:
                       0F4B  ;
                       0F4B  ;       R0 indicates that a response was received from
                       0F4B  ;               the node vs. connection failed
                       0F4B  ;       R1 is status received from the node.
                       0F4B  ;
                       0F4B  ; SIDE EFFECTS:
                       0F4B  ;
                       0F4B  ;       CDRP$L_VAL1-CDRP$L_VAL5 are used for this routine's context.
                       0F4B  ;
                       0F4B  ;--
                       0F4B
                       0F4B  DESCRIBE_NODE:
         F0B2'    30   0F4B          BSBW    CNX$INIT_CDRP                   ; Initialize the CDRP
   54    64 A3    D0   0F4E          MOVL    CSB$L_CLUB(R3),R4               ; Address of CLUB
38 A5       52    D0   0F52          MOVL    R2,CDRP$L_VAL4(R5)              ; Node to describe
   3C A5        8ED0   0F56          POPL    CDRP$L_VAL5(R5)                 ; Return address
   50       05    9A   0F5A          MOVZBL  #CLMCNX$K_FNC_DESC,R0           ; Message code
         03B2     30   0F5D          BSBW    INIT_STD_MSG                    ; Do standard message initialization
20 A5       01    D0   0F60          MOVL    #1,CDRP$L_RSPID(R5)             ; Indicate that response is needed
4C A5     87'AF   9E   0F64          MOVAB   B^BLD_DESC_MSG, -               ; Address of routine to build
                       0F69                  CDRP$L_MSGBLD(R5)               ;  status message
         F094'    30   0F69          BSBW    CNX$SEND_MSG_CSB                ; Send message
                       0F6C  ;
                       0F6C  ;       We are resumed here when the response message arrives.
                       0F6C  ;       Registers contain:
                       0F6C  ;               R0:     Status
                       0F6C  ;               R2:     Address of message buffer
                       0F6C  ;               R3:     Address of CSB
                       0F6C  ;               R4:     Address of PDT
```

```
                    0F6C  ;              R5:     Address of CDRP
                    0F6C  ;
         11 50  E9  0F6C       BLBC    R0,10$                     ; Error sending message
    7E   12 A2  9A  0F6F       MOVZBL  CLMCNX$B_ACK(R2),-(SP)     ; Response from remote node
            50  DD  0F73       PUSHL   R0                         ; Save return status
         F088'  30  0F75       BSBW    CNX$DEALL_MSG_BUF_CSB      ; Release message buffer
                    0F78       DEALLOC_RSPID                      ; Deallocate response ID
            03  BA  0F7E       POPR    #^M<R0,R1>                 ; Restore status and return ACK/NAK
    52   38 A5  D0  0F80  10$: MOVL    CDRP$L_VAL4(R5),R2         ; Address of described CSB
         3C B5  17  0F84       JMP     @CDRP$L_VAL5(R5)           ; Return address
```

```
                              0F87          .SBTTL  BLD_DESC_MSG - Build Message describing a node
                              0F87
                              0F87  ;++
                              0F87  ;
                              0F87  ; FUNCTIONAL DESCRIPTION:
                              0F87  ;
                              0F87  ;       This routine builds a message describing a node.  It uses the
                              0F87  ;       standard message building routine to build the message header.
                              0F87  ;
                              0F87  ; CALLING SEQUENCE:
                              0F87  ;
                              0F87  ;       JSB     BLD_DESC_MSG
                              0F87  ;       IPL is IPL$_SCS
                              0F87  ;
                              0F87  ; INPUT PARAMETERS:
                              0F87  ;
                              0F87  ;       R2:     Address of message buffer
                              0F87  ;       R3:     Address of CSB
                              0F87  ;       R4:     Address of PDT
                              0F87  ;       R5:     Address of CDRP
                              0F87  ;               CDRP$L_VAL1(R5): Byte 0 contains facility code (negated for response
                              0F87  ;                                Byte 1 contains facility specific function code
                              0F87  ;               CDRP$L_VAL2(R5): Transition ID (from CLUB$L_CUR_XTN)
                              0F87  ;               CDRP$L_VAL3(R5): Byte 0 contains transition phase (from CLUB$B_CUR_P
                              0F87  ;                                Byte 1 contains transition code (from CLUB$B_XTN_CO
                              0F87  ;                                Byte 2 contains success/failure flag
                              0F87  ;                                Byte 3 contains reply code
                              0F87  ;               CDRP$L_VAL4(R5): Address of CSB to describe
                              0F87  ;
                              0F87  ; OUTPUT PARAMETERS:
                              0F87  ;
                              0F87  ;       NONE
                              0F87  ;
                              0F87  ; COMPLETION CODES:
                              0F87  ;
                              0F87  ;       NONE
                              0F87  ;
                              0F87  ; SIDE EFFECTS:
                              0F87  ;
                              0F87  ;       R0 and R1 are destroyed
                              0F87  ;
                              0F87  ;--
                              0F87
                              0F87  BLD_DESC_MSG:
              03AF      30    0F87          BSBW    BLD_STD_MSG             ; Fill in standard fields
         50   38 A5     D0    0F8A          MOVL    CDRP$L_VAL4(R5),R0      ; Address of CSB to describe
         51   68 A0     D0    0F8E          MOVL    CSB$L_SB(R0),R1         ; Address of SB of described system
                3F      BB    0F92          PUSHR   #^M<R0,R1,R2,R3,R4,R5>  ; Save registers
 14 A2   18 A1   06     28    0F94          MOVC3   #CLMNOD$S_SYSTEMID, -   ; Fill in described system ID
                              0F9A                  SB$B_SYSTEMID(R1), -
                              0F9A                  CLMNOD$B_SYSTEMID(R2)
                3F      BA    0F9A          POPR    #^M<R0,R1,R2,R3,R4,R5>  ; Restore registers
    1C A2   2C A1       7D    0F9C          MOVQ    SB$Q_SWINCARN(R1), -    ; Fill in described system
                              0FA1                  CLMNOD$Q_SWINCARN(R2)   ;   software incarnation number
    24 A2   4C A0       D0    0FA1          MOVL    CSB$L_CSID(R0), -       ; Described system CSID
                              0FA6                  CLMNOD$L_CSID(R2)
                        05    0FA6          RSB
```

```
                            0FA7          .SBTTL  CNX$RCVD_DESC  - Node description received
                            0FA7
                            0FA7  ;++
                            0FA7  ;
                            0FA7  ; FUNCTIONAL DESCRIPTION:
                            0FA7  ;
                            0FA7  ;       This routine is called when a node description message is received.
                            0FA7  ;       A response must be sent which ACKs or NAKs the request.  The
                            0FA7  ;       request is NAKd if this node does not have a connection.
                            0FA7  ;
                            0FA7  ; CALLING SEQUENCE:
                            0FA7  ;
                            0FA7  ;       JSB     CNX$RCVD_DESC
                            0FA7  ;       IPL is IPL$_SCS
                            0FA7  ;
                            0FA7  ; INPUT PARAMETERS:
                            0FA7  ;
                            0FA7  ;       R2:     Message address
                            0FA7  ;       R3:     CSB of sending system
                            0FA7  ;       R4:     PDT address
                            0FA7  ;       R5:     CDRP address (uninitialized)
                            0FA7  ;
                            0FA7  ; OUTPUT PARAMETERS:
                            0FA7  ;
                            0FA7  ;       NONE
                            0FA7  ;
                            0FA7  ; COMPLETION CODES:
                            0FA7  ;
                            0FA7  ;       NONE
                            0FA7  ;
                            0FA7  ; SIDE EFFECTS:
                            0FA7  ;
                            0FA7  ;       R0-R5 may be destroyed.
                            0FA7  ;--
                            0FA7
                            0FA7  CNX$RCVD_DESC::
                  2C   BB   0FA7          PUSHR   #^M<R2,R3,R5>             ; Save needed registers
               F054'   30   0FA9          BSBW    CNX$INIT_CDRP             ; Initialize the CDRP for the response
            52    6E   D0   0FAC          MOVL    (SP),R2                   ; Restore message address
                0397   30   0FAF          BSBW    MSG_CHECK                 ; Validate message
                  55   D4   0FB2          CLRL    R5                        ; Result CSB address
               F049'   30   0FB4          BSBW    CNX$SCAN_CSBS             ; Iterate over all CSBs
               27 50   E9   0FB7          BLBC    R0,5$                     ; Branch when done
          50   68 A3   D0   0FBA          MOVL    CSB$L_SB(R3),R0           ; Address of System Block
                  0C   BB   0FBE          PUSHR   #^M<R2,R3>                ; Save message buffer and CSB address
14 A2   18 A0     06   29   0FC0          CMPC3   #SB$S_SYSTEMID, -         ; Compare System IDs
                            0FC6                  SB$B_SYSTEMID(R0), -      ;   between SB and
                            0FC6                  CLMNOD$B_SYSTEMID(R2)     ;   incoming message
                  0C   BA   0FC6          POPR    #^M<R2,R3>                ; Restore message buffer and CSB address
                  16   12   0FC8          BNEQ    2$                        ; No match
                  0C   BB   0FCA          PUSHR   #^M<R2,R3>                ; Save message buffer and CSB address
          50   68 A3   D0   0FCC          MOVL    CSB$L_SB(R3),R0           ; Address of System Block
                            0FD0  ;
                            0FD0  ; Use SWINCARN out of SB because local node's CSB doesn't contain a valid
                            0FD0  ; SWINCARN -- it was created and intialized before SWINCARN for the local node
                            0FD0  ; became meaningful.
                            0FD0  ;
```

```
 1C A2    2C A0     08     29  0FD0            CMPC3   #SB$S_SWINCARN, -           ; Compare software incarnation number
                               0FD6                    SB$Q_SWINCARN(R0), -        ;   between CSB and
                               0FD6                    CLMNOD$Q_SWINCARN(R2)       ;   incoming message
                    0C     BA  0FD6            POPR    #^M<R2,R3>                  ; Restore message buffer and CSB address
                    06     12  0FD8            BNEQ    2$                          ; No match
            55      53     D0  0FDA            MOVL    R3,R5                       ; Store result
                 F020'     31  0FDD            BRW     CNX$SCAN_CSBS_EXIT          ; Exit iteration with result
                               0FE0
                           05  0FE0   2$:      RSB                                 ; Return and iterate
                               0FE1
            53      55     D0  0FE1   5$:      MOVL    R5,R3                       ; CSB address of described node
                    29     13  0FE4            BEQL    25$                         ; Described node not found
                    18     E0  0FE6            BBS     #CSB$V_LOCAL,-              ; Branch if local CSB
           0A 60 A3            0FE8                    CSB$L_STATUS(R3),10$
      17 60 A3      19     E1  0FEB            BBC     #CSB$V_STATUS_RCVD, -       ; Branch if no status received
                               0FF0                    CSB$L_STATUS(R3),20$
      12 60 A3      00     E0  0FF0            BBS     #CSB$V_LONG_BREAK, -        ; Branch if long break seen
                               0FF5                    CSB$L_STATUS(R3),20$        ;   and NAK request
 60 A3    00020000 8F      C8  0FF5   10$:     BISL2   #CSB$M_SELECTED, -          ; Mark node SELECTED
                               0FFD                    CSB$L_STATUS(R3)
      4C A3    24 A2       D0  0FFD            MOVL    CLMNOD$L_CSID(R2), -        ; Save node identification
                               1002                    CSB$L_CSID(R3)              ;   for described node
            50      01     D0  1002            MOVL    #1,R0                       ; ACK node description
                    0A     11  1005            BRB     30$
                               1007   ;
                               1007   ; The described node is unknown or no valid connection exists.
                               1007   ; Return a NAK.
                               1007   ;
 60 A3    00020000 8F      CA  1007   20$:     BICL2   #CSB$M_SELECTED, -          ; Mark node deSELECTED
                               100F                    CSB$L_STATUS(R3)
                    50     D4  100F   25$:     CLRL    R0                          ; NAK node description
                    2C     BA  1011   30$:     POPR    #^M<R2,R3,R5>               ; Restore needed registers
                    50     DD  1013            PUSHL   R0                          ; Save status
            50      05     9A  1015            MOVZBL  #CLMCNX$K_FNC_DESC,R0       ; Facility specific function code
                  02EE     30  1018            BSBW    CNX$INIT_STD_RESP           ; Init CDRP for standard response
         36 A5      8E     F6  101B            CVTLB   (SP)+,CDRP$L_VAL3+2(R5)     ; Store success/failure flag
                 EFDE'     31  101F            BRW     CNX$RESP_FORGET             ; Send response message and forget it
```

```
                              1022          .SBTTL  BLD_TOPOLOGY_MSG - Build Message Describing Cluster Topology
                              1022
                              1022  ;++
                              1022  ;
                              1022  ; FUNCTIONAL DESCRIPTION:
                              1022  ;
                              1022  ;       This routine builds a message describing the connectivity of the cluster
                              1022  ;       as known to the local node.  The description is optimistic in that some
                              1022  ;       connections that are reported as good may in fact be bad.  The oppositve
                              1022  ;       cannot occur.  Information about the quorum disk is not passed in this
                              1022  ;       message.
                              1022  ;
                              1022  ; CALLING SEQUENCE:
                              1022  ;
                              1022  ;       JSB     BLD_TOPOLOGY_MSG
                              1022  ;       IPL is IPL$_SCS
                              1022  ;
                              1022  ; INPUT PARAMETERS:
                              1022  ;
                              1022  ;       R2:     Address of message buffer
                              1022  ;       R3:     Address of CSB
                              1022  ;       R4:     Address of PDT
                              1022  ;       R5:     Address of CDRP
                              1022  ;               CDRP$L_VAL1(R5): Byte 0 contains facility code (negated for response
                              1022  ;                                Byte 1 contains facility specific function code
                              1022  ;               CDRP$L_VAL2(R5): Transition ID (from CLUB$L_CUR_XTN)
                              1022  ;               CDRP$L_VAL3(R5): Byte 0 contains transition phase (from CLUB$B_CUR_P
                              1022  ;                                Byte 1 contains transition code (from CLUB$B_XTN_CO
                              1022  ;                                Byte 2 contains success/failure flag
                              1022  ;                                Byte 3 contains reply code
                              1022  ;
                              1022  ; OUTPUT PARAMETERS:
                              1022  ;
                              1022  ;       NONE
                              1022  ;
                              1022  ; COMPLETION CODES:
                              1022  ;
                              1022  ;       NONE
                              1022  ;
                              1022  ; SIDE EFFECTS:
                              1022  ;
                              1022  ;       R0 and R1 are destroyed
                              1022  ;
                              1022  ;--
                              1022
                              1022  BLD_TOPOLOGY_MSG:
                 0314   30    1022          BSBW    BLD_STD_MSG                     ; Fill in standard fields
            50   64 A3  D0    1025          MOVL    CSB$L_CLUB(R3),R0               ; Get CLUB address
            51   10 A0  D0    1029          MOVL    CLUB$L_LOCAL_CSB(R0),R1         ; Local CSB Address
                   3C   BB    102D          PUSHR   #^M<R2,R3,R4,R5>                ; Save registers
  14 A2   008C C1      20   28    102F          MOVC3   #CLMTOP$S_NODEMAP, -            ; Copy data into message buffer
                              1036                  CSB$B_NODEMAP(R1), -
                              1036                  CLMTOP$B_NODEMAP(R2)
                   3C   BA    1036          POPR    #^M<R2,R3,R4,R5>                ; Restore registers
                        05    1038          RSB                                     ; Return and iterate
```

```
                                  1039         .SBTTL  CNX$RCVD_TOPOLOGY  - Topology Request Message Received
                                  1039
                                  1039  ;++
                                  1039  ;
                                  1039  ; FUNCTIONAL DESCRIPTION:
                                  1039  ;
                                  1039  ;       This routine is called when a topology request message is received.
                                  1039  ;       A response must be sent describes the connectivity seen by the
                                  1039  ;       receiving node.  The message itself is used to update the receiving
                                  1039  ;       nodes' understanding of the sending nodes' connectivity.
                                  1039  ;
                                  1039  ; CALLING SEQUENCE:
                                  1039  ;
                                  1039  ;       JSB     CNX$RCVD_TOPOLOGY
                                  1039  ;       IPL is IPL$_SCS
                                  1039  ;
                                  1039  ; INPUT PARAMETERS:
                                  1039  ;
                                  1039  ;       R2:     Message address
                                  1039  ;       R3:     CSB of sending system
                                  1039  ;       R4:     PDT address
                                  1039  ;       R5:     CDRP address (uninitialized)
                                  1039  ;
                                  1039  ; OUTPUT PARAMETERS:
                                  1039  ;
                                  1039  ;       NONE
                                  1039  ;
                                  1039  ; COMPLETION CODES:
                                  1039  ;
                                  1039  ;       NONE
                                  1039  ;
                                  1039  ; SIDE EFFECTS:
                                  1039  ;
                                  1039  ;       R0-R5 may be destroyed.
                                  1039  ;--
                                  1039
                                  1039  CNX$RCVD_TOPOLOGY::
                      2C   BB     1039         PUSHR   #^M<R2,R3,R5>             ; Save needed registers
                   EFC2'   30     103B         BSBW    CNX$INIT_CDRP             ; Initialize the CDRP for the response
               52     6E   7D     103E         MOVQ    (SP),R2                   ; Restore message address and CSB address
                    0305   30     1041         BSBW    MSG_CHECK                 ; Validate message
  008C C3   14 A2     20   28     1044         MOVC3   #CSB$S_NODEMAP, -         ; Fill in connectivity in sending node's
                                  104B                 CLMTOP$B_NODEMAP(R2), -   ;   CSB
                                  104B                 CSB$B_NODEMAP(R3)
                      2C   BA     104B         POPR    #^M<R2,R3,R5>             ; Restore needed registers
               54  64 A3   D0     104D         MOVL    CSB$L_CLUB(R3),R4         ; Fetch CLUB address
               50     0F   9A     1051         MOVZBL  #CLMCNX$K_FNC_TOPOLOGY,R0        ; Facility specific function code
                    02B2   30     1054         BSBW    CNX$INIT_STD_RESP         ; Init CDRP for standard response
        4C A5      C8 AF   9E     1057         MOVAB   B^BLD_TOPOLOGY_MSG, -     ; Store message build routine
                                  105C                 CDRP$L_MSGBLD(R5)
                   EFA1'   31     105C         BRW     CNX$RESP_FORGET           ; Send response message and forget it
```

```
                        105F            .SBTTL  SEND_PH1 - Send Phase 1 Messages
                        105F
                        105F    ;++
                        105F    ;
                        105F    ; FUNCTIONAL DESCRIPTION:
                        105F    ;
                        105F    ;       Send a Phase 1 message to all locked and selected
                        105F    ;       nodes.  A response is required from every node.  If any response
                        105F    ;       is a NAK or any connection breaks, the transition is abandoned
                        105F    ;       without returning to the caller.
                        105F    ;
                        105F    ; CALLING SEQUENCE:
                        105F    ;
                        105F    ;       JSB     SEND_PH1
                        105F    ;       IPL is IPL$_SCS
                        105F    ;
                        105F    ; INPUT PARAMETERS:
                        105F    ;
                        105F    ;       R4 is address of CLUB
                        105F    ;               CLUB$B_CUR_CODE gives transaction type
                        105F    ;               CLUB$L_CTX0 contains R0 argument for INIT_STD_MSG
                        105F    ;               CLUB$L_CTX1 contains message build routine address
                        105F    ;
                        105F    ; OUTPUT PARAMETERS:
                        105F    ;
                        105F    ;       R4 is address of CLUB
                        105F    ;
                        105F    ; COMPLETION CODES:
                        105F    ;
                        105F    ;       NONE
                        105F    ;
                        105F    ; SIDE EFFECTS:
                        105F    ;
                        105F    ;       CLUB$L_RET1 is destroyed.
                        105F    ;       R0-R3, R5 are destroyed.
                        105F    ;--
                        105F
                        105F    SEND_PH1:
          74 A4 8ED0    105F            POPL    CLUB$L_RET1(R4)         ; Save return address
          5A A4   B4    1063            CLRW    CLUB$W_MSGCNT(R4)       ; Initialize waiting count
   59 A4  40 8F   90    1066            MOVB    #CLMCNX$K_PH1, -        ; Mark this as Phase 1
                        106B                    CLUB$B_CUR_PHASE(R4)
         EF92'    30    106B            BSBW    CNX$SCAN_CSBS           ; Iterate over all CSBs
          41 50   E9    106E            BLBC    R0,40$                  ; Branch when done
 1B 60 A3    11   E1    1071            BBC     #CSB$V_SELECTED, -      ; Branch if not selected
                        1076                    CSB$L_STATUS(R3),10$
 16 60 A3    00   E0    1076            BBS     #CSB$V_LONG_BREAK, -    ; Branch if permanent break
                        107B                    CSB$L_STATUS(R3),10$    ;   in connection
 11 60 A3    18   E0    107B            BBS     #CSB$V_LOCAL, -         ; Branch if local CSB
                        1080                    CSB$L_STATUS(R3),10$
         EF7D'    30    1080            BSBW    CNX$ALLOC_WARMCDRP_CSB
         EF7A'    30    1083            BSBW    CNX$RESOURCE_CHECK      ; Watch out for exhausted resources
          09 50   E9    1086            BLBC    R0,20$                  ; No memory available
          5A A4   B6    1089            INCW    CLUB$W_MSGCNT(R4)       ; Include in wait count
             07   10    108C            BSBB    30$                     ; Describe nodes
         EF6F'    31    108E            BRW     CNX$SCAN_CSBS_FORK      ; Fork and resume scan
                        1091
```

```
                      05  1091  10$:    RSB                             ; Return and continue scan
                          1092
               EF6B'  31  1092  20$:    BRW     CNX$SCAN_CSBS_RETRY     ; Delay and resume scan
                          1095
                          1095  ;
                          1095  ; Send proposal to one node
                          1095  ;
      50       70 A4  D0  1095  30$:    MOVL    CLUB$L_CTX0(R4),R0      ; Facility specific message code
                0276  30  1099          BSBW    INIT_STD_MSG            ; Standard CDRP message initialization
   4C A5       78 A4  D0  109C          MOVL    CLUB$L_CTX1(R4), -      ; Address of routine to build
                          10A1                  CDRP$L_MSGBLD(R5)       ;  status message
               EF5C'  30  10A1          BSBW    CNX$SEND_MSG_CSB        ; Send message
                          10A4  ;
                          10A4  ;       We are resumed here when the response message arrives.
                          10A4  ;       Registers contain:
                          10A4  ;               R0:     Status
                          10A4  ;               R2:     Address of message buffer
                          10A4  ;               R3:     Address of CSB
                          10A4  ;               R4:     Address of PDT
                          10A4  ;               R5:     Address of CDRP
                          10A4  ;
                023F  30  10A4          BSBW    CNX$PROCESS_RESPONSE    ; Deallocate storage, get status
               08 50  E8  10A7          BLBS    R0,40$                  ; Branch if ACK
 1C A4   00020000 8F  C8  10AA          BISL2   #CLUB$M_UNLOCK, -       ; Request that all be unlocked
                          10B2                  CLUB$L_FLAGS(R4)
                          10B2  ;
                          10B2  ; terminate threads -- except for the last one
                          10B2  ;
               5A A4  B7  10B2  40$:    DECW    CLUB$W_MSGCNT(R4)
                  01  19  10B5          BLSS    50$                     ; Branch when done
                  05      10B7          RSB                             ; Terminate thread
                          10B8
                0179  30  10B8  50$:    BSBW    CNX$CHECK_UNLOCK        ; Check unlock request bit
               74 B4  17  10BB          JMP     @CLUB$L_RET1(R4)        ; Return to caller
```

```
                           10BE         .SBTTL  SEND_PH2 - Send a Phase 2 Notification
                           10BE
                           10BE ;++
                           10BE ;
                           10BE ; FUNCTIONAL DESCRIPTION:
                           10BE ;
                           10BE ;       Send a Phase 2 notification to all locked and selected
                           10BE ;       nodes.  All messages are sent in parallel.  Breaking
                           10BE ;       connections are ignored.
                           10BE ;
                           10BE ; CALLING SEQUENCE:
                           10BE ;
                           10BE ;       JSB     SEND_PH2
                           10BE ;       IPL is IPL$_SCS
                           10BE ;
                           10BE ; INPUT PARAMETERS:
                           10BE ;
                           10BE ;       CLUB$B_CUR_CODE gives transaction type
                           10BE ;
                           10BE ; OUTPUT PARAMETERS:
                           10BE ;
                           10BE ;       R4 is address of CLUB
                           10BE ;
                           10BE ; COMPLETION CODES:
                           10BE ;
                           10BE ;       NONE
                           10BE ;
                           10BE ; SIDE EFFECTS:
                           10BE ;
                           10BE ;--
                           10BE
                           10BE SEND_PH2:
 59 A4    60 8F      90    10BE         MOVB    #CLMCNX$K_PH2, -          ; Set current phase
                           10C3                 CLUB$B_CUR_PHASE(R4)      ;   into CLUB
           EF3A'     30    10C3         BSBW    CNX$SCAN_CSBS             ; Iterate over all CSBS
           3F 50     E9    10C6         BLBC    R0,DO_PHASE2              ; Branch when done
 1F 60 A3     11     E1    10C9         BBC     #CSB$V_SELECTED, -        ; Branch if not selected
                           10CE                 CSB$L_STATUS(R3),10$
 1A 60 A3     18     E0    10CE         BBS     #CSB$V_LOCAL, -           ; Branch if local CSB
                           10D3                 CSB$L_STATUS(R3),10$
 15 60 A3     00     E0    10D3         BBS     #CSB$V_LONG_BREAK, -      ; Branch if permanent break
                           10D8                 CSB$L_STATUS(R3),10$      ;   in connection
           EF25'     30    10D8         BSBW    CNX$ALLOC_CDRP_ONLY
           EF22'     30    10DB         BSBW    CNX$RESOURCE_CHECK        ; Watch out for exhausted resources
           0D 50     E9    10DE         BLBC    R0,20$                    ; No memory available
                           10E1 ;
                           10E1 ; Send Phase 2 to one node
                           10E1 ;
        50    0A     9A    10E1         MOVZBL  #CLMCNX$K_FNC_PH2,R0      ; Facility specific function code
           022B      30    10E4         BSBW    INIT_STD_MSG              ; Setup CDRP for sending message
           EF16'     30    10E7         BSBW    CNX$SEND_FORGET           ; Send message and forget it
           EF13'     31    10EA         BRW     CNX$SCAN_CSBS_FORK        ; Fork and resume scan
                           10ED
                     05    10ED 10$:    RSB                               ; Return and continue scan
                           10EE
           EF0F'     31    10EE 20$:    BRW     CNX$SCAN_CSBS_RETRY       ; Delay and resume scan
```

```
                          10F1          .SBTTL  CNX$RCVD_PH2 - Phase 2 request received
                          10F1
                          10F1   ;++
                          10F1   ;
                          10F1   ; FUNCTIONAL DESCRIPTION:
                          10F1   ;
                          10F1   ;       This routine is called when a phase 2 request message is received.
                          10F1   ;       The transition inprogress is committed.
                          10F1   ;       No response or immediate acknowledgement is sent.
                          10F1   ;
                          10F1   ; CALLING SEQUENCE:
                          10F1   ;
                          10F1   ;       JSB     CNX$RCVD_PH2
                          10F1   ;       IPL is IPL$_SCS
                          10F1   ;
                          10F1   ; INPUT PARAMETERS:
                          10F1   ;
                          10F1   ;       R2:     Message address
                          10F1   ;       R3:     CSB of sending system
                          10F1   ;       R4:     PDT address
                          10F1   ;
                          10F1   ; OUTPUT PARAMETERS:
                          10F1   ;
                          10F1   ;       NONE
                          10F1   ;
                          10F1   ; COMPLETION CODES:
                          10F1   ;
                          10F1   ;       NONE
                          10F1   ;
                          10F1   ; SIDE EFFECTS:
                          10F1   ;
                          10F1   ;       R0-R5 may be destroyed.
                          10F1   ;--
                          10F1
                          10F1   CNX$RCVD_PH2::
     54    64 A3    D0    10F1          MOVL    CSB$L_CLUB(R3),R4          ; Address of CLUB
07 1C A4      1D    E1    10F5          BBC     #CLUB$V_TRANSITION, -      ; Branch if not locked and
                          10FA                  CLUB$L_FLAGS(R4),10$       ;    bugcheck
  48 A4    0C A2    D1    10FA          CMPL    CLMCNX$L_XTN_ID(R2), -     ; Validate transaction ID
                          10FF                  CLUB$L_CUR_XTN(R4)
           04       13    10FF          BEQL    20$                        ; Branch on match
                          1101   10$:   BUG_CHECK       CNXMGRERR,FATAL    ; Do something more sophisticated later?
                          1105
         EEF8'      30    1105   20$:   BSBW    CNX$DEALL_MSG_BUF_CSB      ; Deallocate message buffer
                          1108
                          1108   ;
                          1108   ; Perform general transition completion operations
                          1108   ;
                          1108   DO_PHASE2:
  34 A4    48 A4    D0    1108          MOVL    CLUB$L_CUR_XTN(R4), -      ; Copy current transaction info into
                          110D                  CLUB$L_LST_XTN(R4)
  38 A4    4C A4    D0    110D          MOVL    CLUB$L_CUR_COORD(R4), -
                          1112                  CLUB$L_LST_COORD(R4)
  3C A4    50 A4    7D    1112          MOVQ    CLUB$Q_CUR_TIME(R4), -
                          1117                  CLUB$Q_LST_TIME(R4)
  44 A4    58 A4    90    1117          MOVB    CLUB$B_CUR_CODE(R4), -
                          111C                  CLUB$B_LST_CODE(R4)
```

```
45 A4   59 A4   90  111C          MOVB      CLUB$B_CUR_PHASE(R4), -
                     1121                    CLUB$B_LST_PHASE(R4)
66 A4   64 A4   B0  1121          MOVW      CLUB$W_NEXT_CSID(R4), - ; Update CSID allocation context
                     1126                    CLUB$W_FIRST_INDEX(R4)
                     1126     ;
                     1126     ; Case on transaction type and branch to appropriate place
                     1126     ;
                     1126          DISPATCH         CLUB$B_CUR_CODE(R4),TYPE=B,PREFIX=CLMCNX$K_XTN_, -
                     1126                    < -
                     1126                    <FORM,FORM_FINISH>, -     ; Cluster formation complete
                     1126                    <JOIN,JOIN_FINISH>, -     ; Joining cluster complete
                     1126                    <RECONFIG,RECONFIG_FINISH>, -     ; Failover complete
                     1126                    <QUORUM,QUORUM_FINISH>, - ; Quorum adjustment complete
                     1126                    >
                     1133          BUG_CHECK        CNXMGRERR,FATAL ; Invalid transaction code
                     1137
```

```
                                1137          .SBTTL  LOCK_NODES - Obtain Coordinator Lock on Selected Nodes
                                1137
                                1137  ;++
                                1137  ;
                                1137  ; FUNCTIONAL DESCRIPTION:
                                1137  ;
                                1137  ;       This routine tries to obtain the coordinator lock on the
                                1137  ;       selected nodes.
                                1137  ;       The attempt fails if any node rejects the request or
                                1137  ;       if the connection to a candidate node breaks.
                                1137  ;       An immediate return is made to the caller's caller.
                                1137  ;       If the locking operation is complete, a return is
                                1137  ;       made to the caller.
                                1137  ;       If the locking operation fails, the transition is aborted
                                1137  ;       and no return is made; rather control passes to the unlock
                                1137  ;       routine.
                                1137  ;
                                1137  ; CALLING SEQUENCE:
                                1137  ;
                                1137  ;       JSB     LOCK_NODES
                                1137  ;       IPL is IPL$_SCS
                                1137  ;
                                1137  ; INPUT PARAMETERS:
                                1137  ;
                                1137  ;       CLUB$B_CUR_CODE gives transaction type
                                1137  ;
                                1137  ; OUTPUT PARAMETERS:
                                1137  ;
                                1137  ;       NONE
                                1137  ;
                                1137  ; COMPLETION CODES:
                                1137  ;
                                1137  ;       NONE
                                1137  ;
                                1137  ; SIDE EFFECTS:
                                1137  ;
                                1137  ;       If CSB is locked, the CSB$M_LOCKED bit in CSB$L_STATUS is set
                                1137  ;       and the CSB$B_REF_CNT is incremented.
                                1137  ;       R0-R5 are destroyed
                                1137  ;
                                1137  ;--
                                1137
                                1137  LOCK_NODES:
 54    00000000'GF     D0       1137          MOVL    G^CLU$GL_CLUB,R4            ; Address of CLUB
            74 A4    8ED0       113E          POPL    CLUB$L_RET1(R4)             ; Save return address
     53     10 A4      D0       1142          MOVL    CLUB$L_LOCAL_CSB(R4),R3     ; Address of local CSB
             EEB7'     30       1146          BSBW    CNX$MARK_LOCKED             ; Mark local CSB locked first
             EEB4'     30       1149          BSBW    CNX$CLUB_FORK               ; Release control briefly
         53    64      DE       114C          MOVAL   CLUB$L_CSBQFL(R4),R3        ; Point to head of CSB list
         53    63      D0       114F  10$:    MOVL    CSB$L_SYSQFL(R3),R3         ; Advance to next CSB
         54    53      D1       1152          CMPL    R3,R4                       ; Done yet?
               3E      13       1155          BEQL    90$                         ; Branch when done
   36 1C A4    11      E0       1157  20$:    BBS     #CLUB$V_UNLOCK, -           ; Branch if unlock requested
                                115C                  CLUB$L_FLAGS(R4),80$
   EE 60 A3    11      E1       115C          BBC     #CSB$V_SELECTED, -          ; Ignore unselected nodes
                                1161                  CSB$L_STATUS(R3),10$
   E9 60 A3    10      E0       1161          BBS     #CSB$V_LOCKED, -            ; Branch if already locked
```

```
                        1166                        CSB$L_STATUS(R3),10$
27 60 A3    00    E0    1166             BBS        #CSB$V_LONG_BREAK, -      ; Branch if long break has been seen
                        116B                        CSB$L_STATUS(R3),80$      ;   and unlock everything
       EE92'      30    116B             BSBW       CNX$ALLOC_WARMCDRP_CSB
       EE8F'      30    116E             BSBW       CNX$RESOURCE_CHECK        ; Watch out for exhausted resources
       05 50      E8    1171             BLBS       R0,30$                    ; Branch if memory available
       EE89'      30    1174             BSBW       CNX$CLUB_WAIT             ; Wait a second, saving R3
          DE      11    1177             BRB        20$                       ; Restart scan
                        1179
    50    03      9A    1179      30$:   MOVZBL     #CLMCNX$K_FNC_LOCK,R0     ; Facility specific function code
        0193      30    117C             BSBW       INIT_STD_MSG              ; Standard CDRP initialization
4C A5  98'AF      9E    117F             MOVAB      B^BLD_LOCK_MSG, -         ; Message building routine
                        1184                        CDRP$L_MSGBLD(R5)
       EE79'      30    1184             BSBW       CNX$SEND_MSG_CSB          ; Send message
                        1187      ;
                        1187      ;             We are resumed here when the response message arrives.
                        1187      ;             Registers contain:
                        1187      ;                     R0:     Status
                        1187      ;                     R2:     Address of message buffer
                        1187      ;                     R3:     Address of CSB
                        1187      ;                     R4:     Address of PDT
                        1187      ;                     R5:     Address of CDRP
                        1187      ;
        015C      30    1187             BSBW       CNX$PROCESS_RESPONSE      ; Deallocate storage, fetch status
       05 50      E9    118A             BLBC       R0,80$                    ; Branch if ACK not received
                        118D      ;
                        118D      ; Setup data to lock node whose CSB is R3
                        118D      ;
       EE70'      30    118D             BSBW       CNX$MARK_LOCKED           ; Mark CSB locked
          BD      11    1190             BRB        10$                       ; Continue with next CSB
                        1192
        00B1      31    1192      80$:   BRW        UNLOCK_ALL                ; Exit from transition
                        1195
       74 B4      17    1195      90$:   JMP        @CLUB$L_RET1(R4)          ; Return to caller's caller
```

```
                        1198        .SBTTL  BLD_LOCK_MSG - Build Coordinator Lock Request Message
                        1198
                        1198  ;++
                        1198  ;
                        1198  ; FUNCTIONAL DESCRIPTION:
                        1198  ;
                        1198  ;       This routine builds a message requesting the coordinator lock.
                        1198  ;       The standard message building routine is used to build the message
                        1198  ;       header.
                        1198  ;
                        1198  ; CALLING SEQUENCE:
                        1198  ;
                        1198  ;       JSB     BLD_LOCK_MSG
                        1198  ;       IPL is IPL$_SCS
                        1198  ;
                        1198  ; INPUT PARAMETERS:
                        1198  ;
                        1198  ;       R2:     Address of message buffer
                        1198  ;       R3:     Address of CSB
                        1198  ;       R4:     Address of PDT
                        1198  ;       R5:     Address of CDRP
                        1198  ;               CDRP$L_VAL1(R5): Byte 0 contains facility code (negated for response
                        1198  ;                                Byte 1 contains facility specific function code
                        1198  ;               CDRP$L_VAL2(R5): Transition ID (from CLUB$L_CUR_XTN)
                        1198  ;               CDRP$L_VAL3(R5): Byte 0 contains transition phase (from CLUB$B_CUR_P
                        1198  ;                                Byte 1 contains transition code (from CLUB$B_XTN_CO
                        1198  ;                                Byte 2 contains success/failure flag
                        1198  ;                                Byte 3 contains reply code
                        1198  ;               CDRP$L_VAL4(R5): Address of CSB to describe
                        1198  ;
                        1198  ; OUTPUT PARAMETERS:
                        1198  ;
                        1198  ;       NONE
                        1198  ;
                        1198  ; COMPLETION CODES:
                        1198  ;
                        1198  ;       NONE
                        1198  ;
                        1198  ; SIDE EFFECTS:
                        1198  ;
                        1198  ;       R0 and R1 are destroyed
                        1198  ;
                        1198  ;--
                        1198
                        1198  BLD_LOCK_MSG:
          019E   30     1198        BSBW    BLD_STD_MSG             ; Fill in standard fields
     50   64 A3  D0     119B        MOVL    CSB$L_CLUB(R3),R0       ; Address of CLUB
14 A2     50 A0  7D     119F        MOVQ    CLUB$Q_CUR_TIME(R0), -  ; Fill in transition time-stamp
                        11A4                CLMLCK$Q_XTN_TIME(R2)
                 05     11A4        RSB
```

```
                          11A5          .SBTTL  CNX$RCVD_LOCK - Lock request received
                          11A5
                          11A5  ;++
                          11A5  ;
                          11A5  ; FUNCTIONAL DESCRIPTION:
                          11A5  ;
                          11A5  ;       This routine is called when a lock request message is received.
                          11A5  ;       A response must be sent which ACKs or NAKs the request.  The
                          11A5  ;       request is NAKd if this node is already locked or in transition.
                          11A5  ;
                          11A5  ; CALLING SEQUENCE:
                          11A5  ;
                          11A5  ;       JSB     CNX$RCVD_LOCK
                          11A5  ;       IPL is IPL$_SCS
                          11A5  ;
                          11A5  ; INPUT PARAMETERS:
                          11A5  ;
                          11A5  ;       R2:     Message address
                          11A5  ;       R3:     CSB of sending system
                          11A5  ;       R4:     PDT address
                          11A5  ;       R5:     CDRP address (uninitialized)
                          11A5  ;
                          11A5  ; OUTPUT PARAMETERS:
                          11A5  ;
                          11A5  ;       NONE
                          11A5  ;
                          11A5  ; COMPLETION CODES:
                          11A5  ;
                          11A5  ;       NONE
                          11A5  ;
                          11A5  ; SIDE EFFECTS:
                          11A5  ;
                          11A5  ;       R0-R5 may be destroyed.
                          11A5  ;--
                          11A5
                          11A5  CNX$RCVD_LOCK::
               52    DD   11A5          PUSHL   R2                          ; Save needed register
            EE56'    30   11A7          BSBW    CNX$INIT_CDRP               ; Initialize the CDRP for the response
      54    64 A3    D0   11AA          MOVL    CSB$L_CLUB(R3),R4           ; Address of CLUB
               04    BA   11AE          POPR    #^M<R2>                     ; Restore message address
 50   68 A4 0C A2    C3   11B0          SUBL3   CLMCNX$L_XTN_ID(R2), -      ; Update largest transaction seen -
                          11B6                  CLUB$L_MAX_XTN(R4),R0       ;   status is only interesting result
               65    18   11B6          BGEQ    10$                         ; No update needed
      68 A4 0C A2    D0   11B8          MOVL    CLMCNX$L_XTN_ID(R2), -      ; Update local information
                          11BD                  CLUB$L_MAX_XTN(R4)
   63 1C A4    13    E1   11BD          BBC     #CLUB$V_INIT, -             ; Branch if initialization not
                          11C2                  CLUB$L_FLAGS(R4),20$        ;   complete and NAK request
   5E 1C A4    1D    E2   11C2          BBSS    #CLUB$V_TRANSITION, -       ; Branch if busy and mark us busy
                          11C7                  CLUB$L_FLAGS(R4), 20$
         0084 C4     D4   11C7          CLRL    CLUB$L_TQE(R4)              ; Abandon any timeout in progress
   50    00CC C4     9E   11CB          MOVAB   CLUB$B_FORK_BLOCK(R4),R0 ; Address of transition fork block
   03 1C A0    00    E5   11D0          BBCC    #CLUBFKB$V_FKB_BUSY, -      ; Branch if fork block is not
                          11D5                  CLUBFKB$L_STATUS(R0),5$ ;  busy and mark no busy
      50    60       0F   11D5          REMQUE  CLUBFKB$B_FORK_BLOCK(R0),R0 ; Remove from queue
                          11D8  5$:
                          11D8  ;
                          11D8  ; Initialization for accepting lock request and beginning new transition
```

```
                             11D8  ;
               EE25'   30    11D8            BSBW    CNX$INIT_CSBS           ; Initialize all CSBs for new transition
 1C A4  00020000 8F    CA    11DB            BICL2   #CLUB$M_UNLOCK, -       ; Clear unlock flag
                             11E3                    CLUB$L_FLAGS(R4)
      50    10 A4      D0    11E3            MOVL    CLUB$L_LOCAL_CSB(R4),R0 ; Address of local CSB
 60 A0  00020000 8F    C8    11E7            BISL2   #CSB$M_SELECTED, -      ; Mark local node SELECTED
                             11EF                    CSB$L_STATUS(R0)
 60 A3  00020000 8F    C8    11EF            BISL2   #CSB$M_SELECTED, -      ; Mark coordinator SELECTED
                             11F7                    CSB$L_STATUS(R3)
     48 A4    0C A2    D0    11F7            MOVL    CLMCNX$L_XTN_ID(R2), -  ; Update local information about
                             11FC                    CLUB$L_COR_XTN(R4)      ;   transition
     4C A4    4C A3    D0    11FC            MOVL    CSB$L_CSID(R3), -       ; CSID of coordinator
                             1201                    CLUB$L_CUR_COORD(R4)
     58 A4    11 A2    90    1201            MOVB    CLMCNX$B_XTN_CODE(R2), - ; Transition type
                             1206                    CLUB$B_COR_CODE(R4)
     59 A4    10 A2    90    1206            MOVB    CLMCNX$B_XTN_PHASE(R2), - ; Transition phase
                             120B                    CLUB$B_COR_PHASE(R4)
     50 A4    14 A2    7D    120B            MOVQ    CLMLCK$Q_XTN_TIME(R2), - ; Transition time-stamp
                             1210                    CLUB$Q_COR_TIME(R4)
        5C A4    53    D0    1210            MOVL    R3,CLUB$L_COORD(R4)     ; CSB of coordinator
     64 A4    66 A4    B0    1214            MOVW    CLUB$W_FIRST_INDEX(R4), - ; Initialize CSID allocation context
                             1219                    CLUB$W_NEXT_CSID(R4)
                 01    DD    1219            PUSHL   #1                      ; Accept request
                 0A    11    121B            BRB     30$
                             121D  ;
                             121D  ; Transition number is invalid.
                             121D  ; Lock request will be rejected.
                             121D  ; Request that status be send to remote node.
                             121D  ;
 60 A3  04000000 8F    C8    121D  10$:      BISL2   #CSB$M_SEND_STATUS, -   ; Send status request
                             1225                    CSB$L_STATUS(R3)
                             1225  ;
                             1225  ; Common reject point
                             1225  ;
                 00    DD    1225  20$:      PUSHL   #0                      ; Reject lock request
            50   03    9A    1227  30$:      MOVZBL  #CLMCNX$K_FNC_LOCK,R0   ; Facility specific function code
               00DC    30    122A            BSBW    CNX$INIT_STD_RESP       ; Init CDRP for standard response
         36 A5   8E    F6    122D            CVTLB   (SP)+,CDRP$L_VAL3+2(R5) ; Store success/failure flag
               EDCC'   31    1231            BRW     CNX$RESP_FORGET         ; Send response and forget it
```

```
                         1234          .SBTTL  CNX$CHECK_UNLOCK - Check UNLOCK flag
                         1234
                         1234   ;++
                         1234   ;
                         1234   ; FUNCTIONAL DESCRIPTION:
                         1234   ;
                         1234   ;       Test the UNLOCK flag in the CLUB.  If it is clear, return to the caller.
                         1234   ;       If it is set, remove the caller's PC from the stack, and branch to UNLOCK_AL
                         1234   ;       This will eventually return to the caller's caller.
                         1234   ;
                         1234   ; CALLING SEQUENCE:
                         1234   ;
                         1234   ;       JSB     CNX$CHECK_UNLOCK
                         1234   ;       IPL is IPL$_SCS
                         1234   ;
                         1234   ; INPUT PARAMETERS:
                         1234   ;
                         1234   ;       The CLUB$M_UNLOCK bit CLUB$L_FLAGS is the primary input
                         1234   ;
                         1234   ; OUTPUT PARAMETERS:
                         1234   ;
                         1234   ;       R4 contains the address of the CLUB
                         1234   ;
                         1234   ; COMPLETION CODES:
                         1234   ;
                         1234   ;       NONE
                         1234   ;
                         1234   ; SIDE EFFECTS:
                         1234   ;
                         1234   ;       NONE
                         1234   ;
                         1234   ;--
                         1234
                         1234   CNX$CHECK_UNLOCK::
 54   00000000'GF   D0   1234           MOVL    G^CLU$GL_CLUB,R4            ; Get address of CLUB
    01 1C A4   11   E0   123B           BBS     #CLUB$V_UNLOCK, -           ; Branch if unlock is request and abort
                         1240                   CLUB$L_FLAGS(R4),10$        ;   this transition
                    05   1240           RSB
                         1241
          5E   04   C0   1241   10$:    ADDL2   #4,SP                       ; Remove caller's PC
               00   11   1244           BRB     UNLOCK_ALL                  ; Abort transition and return to caller's ca
```

```
                          1246          .SBTTL  UNLOCK_ALL - Release Coordinator Lock on All Nodes
                          1246
                          1246  ;++
                          1246  ;
                          1246  ; FUNCTIONAL DESCRIPTION:
                          1246  ;
                          1246  ;       This routine releases the coordinator lock on all nodes.
                          1246  ;       A breaking link counts as a successful release of the lock.
                          1246  ;
                          1246  ; CALLING SEQUENCE:
                          1246  ;
                          1246  ;       JSB     UNLOCK_ALL
                          1246  ;       IPL is IPL$_SCS
                          1246  ;
                          1246  ; INPUT PARAMETERS:
                          1246  ;
                          1246  ;       CLUB$B_CUR_CODE gives transaction type
                          1246  ;
                          1246  ; OUTPUT PARAMETERS:
                          1246  ;
                          1246  ;       NONE
                          1246  ;
                          1246  ; COMPLETION CODES:
                          1246  ;
                          1246  ;       NONE
                          1246  ;
                          1246  ; SIDE EFFECTS:
                          1246  ;
                          1246  ;       R0-R5 are destroyed
                          1246  ;
                          1246  ;--
                          1246
                          1246  UNLOCK_ALL:
             EDB7'   30   1246          BSBW    CNX$CHECK_QUORUM        ; Adjust dynamic quorum flag, hang if no quo
54      00000000'GF  D0   1249          MOVL    G^CLU$GL_CLUB,R4        ; Address of CLUB
         53    10 A4 D0   1250          MOVL    CLUB$L_LOCAL_CSB(R4),R3 ; Local CSB address
   10 60 A3    10    E0   1254          BBS     #CSB$V_LOCKED, -        ; Branch if still locked
                          1259                  CSB$L_STATUS(R3),5$
      5C A4    53    D1   1259          CMPL    R3,CLUB$L_COORD(R4)     ; Is local system the coordinator?
               0A    12   125D          BNEQ    5$                      ; Branch if not coordinator
               55    D4   125F          CLRL    R5                      ; No CSB address for message
   50      0000'CF   9E   1261          MOVAB   COMPLETE_MSG,R0         ; Cluster state transition complete message
             ED97'   30   1266          BSBW    CNX$CONFIG_CHANGE       ; Make this state known
                          1269  5$:
   59 A4    50 8F    90   1269          MOVB    #CLMCNX$K_UNLOCK, -     ; Set phase to UNLOCK
                          126E                  CLUB$B_CUR_PHASE(R4)
             ED8F'   30   126E          BSBW    CNX$SCAN_CSBS           ; Iterate over all CSB's
             4E 50   E9   1271          BLBC    R0,UNLOCK_NODE          ; Branch when done
   05 60 A3    01    E0   1274          BBS     #CSB$V_MEMBER, -        ; Branch if a cluster member
                          1279                  CSB$L_STATUS(R3),10$
   00 60 A3    1A    E3   1279          BBCS    #CSB$V_SEND_STATUS, -   ; Request status for non-cluster nodes
                          127E                  CSB$L_STATUS(R3), 10$   ;   informing them of this nodes' new status
   35 60 A3    10    E1   127E  10$:    BBC     #CSB$V_LOCKED, -        ; Branch if not locked
                          1283                  CSB$L_STATUS(R3),40$
   5C A4    10 A4    D1   1283          CMPL    CLUB$L_LOCAL_CSB(R4), - ; Is local system the
                          1288                  CLUB$L_COORD(R4)        ;   coordinator?
               28    12   1288          BNEQ    30$                     ; Branch if not coordinator
```

```
19 60 A3     18    E0  128A          BBS     #CSB$V_LOCAL, -            ; Branch if local CSB
                       128F                  CSB$L_STATUS(R3),20$
1E 60 A3     00    E0  128F          BBS     #CSB$V_LONG_BREAK, -       ; Branch if connection really
                       1294                  CSB$L_STATUS(R3),30$       ;   broken
           ED69'   30  1294          BSBW    CNX$ALLOC_CDRP_ONLY        ; Get a CDRP
           ED66'   30  1297          BSBW    CNX$RESOURCE_CHECK         ; Watch out for exhausted resources
          1C 50    E9  129A          BLBC    R0,50$                     ; No memory available
       50    04    9A  129D          MOVZBL  #CLMCNX$K_FNC_UNLOCK,R0    ; Facility specific function code
           006F    30  12A0          BSBW    INIT_STD_MSG               ; Init CDRP for standard message
           ED5A'   30  12A3          BSBW    CNX$SEND_FORGET            ; Send message and forget it
             0A    11  12A6          BRB     30$                        ; Branch to common exit
                       12A8
                       12A8  ; Output ABORT message only once -- when processing local node
                       12A8  ;
             55    D4  12A8  20$:    CLRL    R5                         ; No CSB address
   50   0000'CF    9E  12AA          MOVAB   UNLOCK_MSG,R0              ; Aborting transition message
           ED4E'   30  12AF          BSBW    CNX$CONFIG_CHANGE          ; Make this state known
                       12B2
                       12B2  ; Common exit
                       12B2  ;
           ED4B'   30  12B2  30$:    BSBW    CNX$MARK_UNLOCKED          ; Mark CSB not locked.  If the CSB
                       12B5                                             ;   is deleted, R3 is previous list entry
           ED48'   31  12B5          BRW     CNX$SCAN_CSBS_FORK         ; Fork and then continue scan
                       12B8
                   05  12B8  40$:    RSB
                       12B9
           ED44'   31  12B9  50$:    BRW     CNX$SCAN_CSBS_RETRY        ; Delay and then resume scan
```

```
                      12BC                .SBTTL  CNX$RCVD_UNLOCK - Unlock request received
                      12BC
                      12BC        ;++
                      12BC        ;
                      12BC        ; FUNCTIONAL DESCRIPTION:
                      12BC        ;
                      12BC        ;       This routine is called when an unlock request message is received.
                      12BC        ;       The node is unlocked. No response or immediate acknowledgement is sent.
                      12BC        ;
                      12BC        ; CALLING SEQUENCE:
                      12BC        ;
                      12BC        ;       JSB     CNX$RCVD_UNLOCK
                      12BC        ;       IPL is IPL$_SCS
                      12BC        ;
                      12BC        ; INPUT PARAMETERS:
                      12BC        ;
                      12BC        ;       R2:     Message address
                      12BC        ;       R3:     CSB of sending system
                      12BC        ;       R4:     PDT address
                      12BC        ;
                      12BC        ; OUTPUT PARAMETERS:
                      12BC        ;
                      12BC        ;       NONE
                      12BC        ;
                      12BC        ; COMPLETION CODES:
                      12BC        ;
                      12BC        ;       NONE
                      12BC        ;
                      12BC        ; SIDE EFFECTS:
                      12BC        ;
                      12BC        ;       R0-R5 may be destroyed.
                      12BC        ;--
                      12BC
                      12BC        CNX$RCVD_UNLOCK::
          008A   30   12BC                BSBW    MSG_CHECK               ; Validate message
          ED3E'  30   12BF                BSBW    CNX$DEALL_MSG_BUF_CSB   ; Deallocate message buffer
                      12C2        ;
                      12C2        ; Right here, all nodes have been unlocked.
                      12C2        ; May need to start up some activity here.
                      12C2        ;
                      12C2        UNLOCK_NODE:
  20 A4    22 A4   B1 12C2                CMPW    CLUB$W_VOTES(R4), -     ; Is there a quorum?
                      12C7                        CLUB$W_QUORUM(R4)
             09    1E 12C7                BGEQU   10$                     ; Branch if yes
  04 1C A4   02    E1 12C9                BBC     #CLUB$V_SHUTDOWN, -     ; Branch if not in shutdown mode
                      12CE                        CLUB$L_FLAGS(R4),10$
                      12CE                BUG_CHECK       OPERATOR,FATAL  ; Finish cluster-wide shutdown
                      12D2
  00 1C A4   1B    E5 12D2        10$:    BBCC    #CLUB$V_ADJ_QUORUM, -   ; Clear quorum adjustment flag
                      12D7                        CLUB$L_FLAGS(R4),20$
     59 A4   10    90 12D7        20$:    MOVB    #CLMCNX$K_IDLE, -       ; Set phase to IDLE
                      12DB                        CLUB$B_CUR_PHASE(R4)
  00 1C A4   1D    E5 12DB                BBCC    #CLUB$V_TRANSITION, -   ; Clear transition flag
                      12E0                        CLUB$L_FLAGS(R4),30$
          ED1D'  30   12E0        30$:    BSBW    CNX$CHECK_QUORUM        ; Check dynamic quorum
          F1D0   31   12E3                BRW     START_TIMEOUT           ; Keep the home fires burning...
```

```
                        12E6          .SBTTL  CNX$PROCESS_RESPONSE - Process simple response message
                        12E6
                        12E6    ;++
                        12E6    ;
                        12E6    ; FUNCTIONAL DESCRIPTION:
                        12E6    ;
                        12E6    ;       Process a reponse message by deallocating the message buffer
                        12E6    ;       and CDRP.
                        12E6    ;       Return status indicating whether the connection broke, the
                        12E6    ;       response was an ACK, or the response was an ACK.
                        12E6    ;
                        12E6    ; CALLING SEQUENCE:
                        12E6    ;
                        12E6    ;       JSB     CNX$PROCESS_RESPONSE
                        12E6    ;       IPL is IPL$_SCS
                        12E6    ;
                        12E6    ; INPUT PARAMETERS:
                        12E6    ;
                        12E6    ;       R0:     Status
                        12E6    ;       R2:     Address of message buffer
                        12E6    ;       R3:     Address of CSB
                        12E6    ;       R4:     Address of PDT
                        12E6    ;       R5:     Address of CDRP
                        12E6    ;
                        12E6    ; OUTPUT PARAMETERS:
                        12E6    ;
                        12E6    ;       R4:     Address of CLUB
                        12E6    ;
                        12E6    ; COMPLETION CODES:
                        12E6    ;
                        12E6    ;       R0:
                        12E6    ;               1 (success) Response was ACK
                        12E6    ;               0 (failure) Response was NAK
                        12E6    ;               2 (failure) Connection broke
                        12E6    ;
                        12E6    ; SIDE EFFECTS:
                        12E6    ;
                        12E6    ;       R1, R2, and R5 are destroyed
                        12E6    ;
                        12E6    ;--
                        12E6
                        12E6    CNX$PROCESS_RESPONSE::
      0B 50       E9    12E6            BLBC    R0,10$                  ; Error sending message
 7E   12 A2       9A    12E9            MOVZBL  CLM[CNX$B_ACK(R2),-(SP) ; Save response
        ED10'     30    12ED            BSBW    CNX$DEAL[_WARM[CDRP_CSB ; Recycle warm CDRP
          01      BA    12F0            POPR    #^M<R0>                 ; Restore response ACK/NAK
          10      11    12F2            BRB     20$                     ; Branch to common exit
                        12F4    ;
          53      DD    12F4    10$:    PUSHL   R3                      ; Save CSB address
      50  55      D0    12F6            MOVL    R5,R0                   ; Address of CDRP
   00000000'GF    16    12F9            JSB     G^EXE$DEANONPAGED       ; Deallocate CDRP
          08      BA    12FF            POPR    #^M<R3>                 ; Restore CSB address
      50  02      D0    1301            MOVL    #2,R0                   ; Return status code
 54   64 A3       D0    1304    20$:    MOVL    CSB$L_CLUB(R3),R4       ; Address of CLUB
          05            1308            RSB                             ; Return to caller
```

```
                        1309          .SBTTL  INIT_STD_MSG - Common CDRP initialization for standard message
                        1309          .SBTTL  CNX$INIT_STD_RESP - Common CDRP initialization for standard response
                        1309
                        1309     ;++
                        1309     ;
                        1309     ; FUNCTIONAL DESCRIPTION:
                        1309     ;
                        1309     ;       This routine initializes most of the CDRP fields used by the standard
                        1309     ;       message build routines.  Two entry points are provided -- one for
                        1309     ;       normal messages and one for responses.
                        1309     ;       These routines are provided for use in conjunction with the BLD_STD_MSG
                        1309     ;       message building routine.
                        1309     ;
                        1309     ; CALLING SEQUENCE:
                        1309     ;
                        1309     ;       JSB     INIT_STD_MSG
                        1309     ;       JSB     CNX$INIT_STD_RESP
                        1309     ;       IPL is IPL$_SCS
                        1309     ;
                        1309     ; INPUT PARAMETERS:
                        1309     ;
                        1309     ;       R0:     Message specified data
                        1309     ;               Byte 0: Facility specific function code
                        1309     ;               Byte 2: ACK byte data (INIT_STD_MSG only)
                        1309     ;               Byte 3: REPLY byte data (INIT_STD_MSG only)
                        1309     ;       R3:     Address of CSB
                        1309     ;       R4:     Address of CLUB
                        1309     ;               CLUB$L_CUR_XTN(R4): Current transition id
                        1309     ;               CLUB$B_CUR_PHASE(R4): Current transition phase
                        1309     ;               CLUB$B_CUR_CODE(R4): Current transition code
                        1309     ;       R5:     Address of CDRP
                        1309     ;
                        1309     ; OUTPUT PARAMETERS:
                        1309     ;
                        1309     ;       NONE
                        1309     ;
                        1309     ; COMPLETION CODES:
                        1309     ;
                        1309     ;       NONE
                        1309     ;
                        1309     ; SIDE EFFECTS:
                        1309     ;
                        1309     ;       R0 and R1 are destroyed
                        1309     ;
                        1309     ;--
                        1309
                        1309          .ENABLE LSB
                        1309
                        1309     CNX$INIT_STD_RESP::
   51    81 8F    90    1309          MOVB    #<CLSMSG$K_FAC_CNX ! -   ; Facility message code
                        130D                   CLSMSG$M_RESPMSG>,R1    ;   with response tag
      50    50    9A    130D          MOVZBL  R0,R0                    ; Ignore high order bits *** temp
            03    11    1310          BRB     10$
                        1312
                        1312     INIT_STD_MSG:
      51    01    90    1312          MOVB    #CLSMSG$K_FAC_CNX,R1     ; Facility message code
         2C A5    D4    1315     10$: CLRL    CDRP$L_VA[1(R5)          ; Initialize other fields
```

```
      2C A5    51   90  1318        MOVB    R1,CDRP$L_VAL1+0(R5)       ; Facility identification code
      2D A5    50   90  131C        MOVB    R0,CDRP$L_VAL1+1(R5)       ; Facility specific code
   30 A5    48 A4   D0  1320        MOVL    CLUB$L_CUR_XTN(R4), -      ; Current transition ID
                        1325                CDRP$L_VAL2(R5)
      34 A5    50   D0  1325        MOVL    R0,CDRP$L_VAL3(R5)         ; Store byte 2 of R0 as ack  field
                        1329                                           ;   and byte 3 of R0 as reply field
   34 A5    59 A4   90  1329        MOVB    CLUB$B_CUR_PHASE(R4), -    ; Current transition phase
                        132E                CDRP$L_VAL3(R5)
   35 A5    58 A4   90  132E        MOVB    CLUB$B_CUR_CODE(R4), -     ; Current transition code
                        1333                CDRP$L_VAL3+1(R5)
   4C A5    39'AF   9E  1333        MOVAB   B^BLD_STD_MSG, -           ; Standard message build routine
                        1338                CDRP$L_MSGBLD(R5)
                    05  1338        RSB
                        1339
                        1339        .DISABLE        LSB
```

```
                             1339          .SBTTL  BLD_STD_MSG - Build Standard Message from CDRP
                             1339
                             1339   ;++
                             1339   ;
                             1339   ; FUNCTIONAL DESCRIPTION:
                             1339   ;
                             1339   ;       This routine is a common message build routine that fills in
                             1339   ;       the common message header from data in the CDRP.
                             1339   ;       There is no dependence upon the contents of the CLUB in building
                             1339   ;       the message.  This means that this routine is safe to use for
                             1339   ;       messages that are queued and forgotten and which can be sent
                             1339   ;       at a later time when the state of the world has changed.
                             1339   ;
                             1339   ; CALLING SEQUENCE:
                             1339   ;
                             1339   ;       JSB     BLD_STD_MSG
                             1339   ;       IPL is IPL$_SCS
                             1339   ;
                             1339   ; INPUT PARAMETERS:
                             1339   ;
                             1339   ;       R2:     Address of message buffer
                             1339   ;       R3:     Address of CSB
                             1339   ;       R4:     Address of PDT
                             1339   ;       R5:     Address of CDRP
                             1339   ;               CDRP$L_VAL1(R5): Byte 0 contains facility code (negated for response
                             1339   ;                                Byte 1 contains facility specific function code
                             1339   ;               CDRP$L_VAL2(R5): Transition ID (from CLUB$L_CUR_XTN)
                             1339   ;               CDRP$L_VAL3(R5): Byte 0 contains transition phase (from CLUB$B_CUR_P
                             1339   ;                                Byte 1 contains transition code (from CLUB$B_XTN_CO
                             1339   ;                                Byte 2 contains success/failure flag
                             1339   ;                                Byte 3 contains reply code
                             1339   ;
                             1339   ; OUTPUT PARAMETERS:
                             1339   ;
                             1339   ;       NONE
                             1339   ;
                             1339   ; COMPLETION CODES:
                             1339   ;
                             1339   ;       NONE
                             1339   ;
                             1339   ; SIDE EFFECTS:
                             1339   ;
                             1339   ;       R0 and R1 are destroyed
                             1339   ;
                             1339   ;--
                             1339
                             1339   BLD_STD_MSG:
 08 A2    2C A5     90       1339           MOVB    CDRP$L_VAL1+0(R5), -     ; Facility message / response code
                             133E                   CLSMSG$B_FACILITY(R2)
 09 A2    2D A5     90       133E           MOVB    CDRP$L_VAL1+1(R5), -     ; Facility specific function code
                             1343                   CLSMSG$B_FUNC(R2)
                             1343           ASSUME  CLMCNX$B_XTN_PHASE EQ CLMCNX$L_XTN_ID+4
                             1343           ASSUME  CDRP$L_VAL3 EQ CDRP$L_VAL2+4
                             1343           ASSUME  CLMCNX$B_XTN_CODE EQ CLMCNX$B_XTN_PHASE+1
                             1343           ASSUME  CLMCNX$B_ACK EQ CLMCNX$B_XTN_CODE+1
                             1343           ASSUME  CLMCNX$B_REPLY EQ CLMCNX$B_ACK+1
 0C A2    30 A5     7D       1343           MOVQ    CDRP$L_VAL2(R5), -       ; Standard information
```

```
      1348  4317              CLMCNX$L_XTN_ID(R2)
  05  1348  4318        RSB
```

```
                              1349        .SBTTL  MSG_CHECK - Standard incoming message verification
                              1349
                              1349  ;++
                              1349  ;
                              1349  ; FUNCTIONAL DESCRIPTION:
                              1349  ;
                              1349  ;       This routine is called to perform standard checks upon an
                              1349  ;       incoming message.  These checks may not be application to
                              1349  ;       messages intitiating or completing a transition.
                              1349  ;
                              1349  ; CALLING SEQUENCE:
                              1349  ;
                              1349  ;       JSB     MSG_CHECK
                              1349  ;       IPL is IPL$_SCS
                              1349  ;
                              1349  ; INPUT PARAMETERS:
                              1349  ;
                              1349  ;       R2:     Message address
                              1349  ;       R3:     CSB of sending system
                              1349  ;       R4:     PDT address
                              1349  ;       R5:     CDRP address (uninitialized)
                              1349  ;
                              1349  ; OUTPUT PARAMETERS:
                              1349  ;
                              1349  ;       R4:     Address of the CLUB
                              1349  ;
                              1349  ; COMPLETION CODES:
                              1349  ;
                              1349  ;       NONE
                              1349  ;
                              1349  ; SIDE EFFECTS:
                              1349  ;
                              1349  ;       R0-R1 may be destroyed.
                              1349  ;--
                              1349
                              1349 MSG_CHECK:
      54     64 A3   D0       1349        MOVL    CSB$L_CLUB(R3),R4           ; Address of CLUB
   48 A4     0C A2   D1       134D        CMPL    CLMCNX$L_XTN_ID(R2), -     ; Check for inconsistency
                              1352                CLUB$L_COR_XTN(R4)
                0B   12       1352        BNEQ    10$                         ; Bugcheck on inconsistency
   06 1C A4     1D   E1       1354        BBC     #CLUB$V_TRANSITION, -      ; Branch on inconsistency
                              1359                CLUB$L_FLAGS(R4), 10$      ;   and bugcheck
   59 A4     10 A2   90       1359        MOVB    CLMCNX$B_XTN_PHASE(R2), - ; Save transition phase
                              135E                CLUB$B_COR_PHASE(R4)
                     05       135E        RSB
                              135F
                              135F 10$:   BUG_CHECK       CNXMGRERR,FATAL ; Inconsistent message
```

```
                              1363          .SBTTL  UPDATE_QUORUM - Update Quorum Parameters in CLUB
                              1363
                              1363  ;++
                              1363  ;
                              1363  ; FUNCTIONAL DESCRIPTION:
                              1363  ;
                              1363  ;       Update the CLUB$W_QUORUM, CLUB$W_VOTES, CLUB$W_QDVOTES, and
                              1363  ;       CLUB$W_NODES fields of the CLUB at the conclusion of a fully
                              1363  ;       synchronized state change.
                              1363  ;
                              1363  ; CALLING SEQUENCE:
                              1363  ;
                              1363  ;       JSB     UPDATE_QUORUM
                              1363  ;       IPL is IPL$_SCS
                              1363  ;
                              1363  ; INPUT PARAMETERS:
                              1363  ;
                              1363  ;       CSB'S in list with CSB$M_SELECTED bit set
                              1363  ;       CLUB$W_NEWQUORUM is quorum
                              1363  ;       CLUB$W_NEWQDVOTES is votes assigned to quorum disk
                              1363  ;       CLUB$V_QF_NEWVOTE is quorum disk membership flag
                              1363  ;
                              1363  ; OUTPUT PARAMETERS:
                              1363  ;
                              1363  ;       R4 is address of CLUB
                              1363  ;       CLUB$W_QUORUM is new quorum
                              1363  ;       CLUB$W_VOTES is new number of votres present in the cluster
                              1363  ;       CLUB$W_NODES is number of nodes in cluster
                              1363  ;       CLUB$V_QF_VOTE indicates that the quorum disk is a member
                              1363  ;
                              1363  ; COMPLETION CODES:
                              1363  ;
                              1363  ;       NONE
                              1363  ;
                              1363  ; SIDE EFFECTS:
                              1363  ;
                              1363  ;       R0 and R1 are destroyed.
                              1363  ;--
                              1363
                              1363  UPDATE_QUORUM:
                     0C   BB  1363          PUSHR   #^M<R2,R3>                 ; Save registers
         54   00000000'GF   D0  1365          MOVL    G^CLU$GL_CLUB,R4           ; Address of CLUB
          20 A4   00A4 C4   B0  136C          MOVW    CLUB$W_NEWQUORUM(R4), -    ; Update cluster quorum
                              1372                  CLUB$W_QUORUM(R4)
          00AE C4   46 A4   B0  1372          MOVW    CLUB$W_NEWQDVOTES(R4), -   ; Update quorum disk votes
                              1378                  CLUB$W_QDVOTES(R4)
                  24 A4   B4  1378          CLRW    CLUB$W_NODES(R4)           ; Initialize count of nodes
                  22 A4   B4  137B          CLRW    CLUB$W_VOTES(R4)           ; Initialize vote count
     1C A4   02000000 8F   CA  137E          BICL    #CLUB$M_QF_VOTE, -         ; Assume no quorum disk membership
                              1386                  CLUB$L_FLAGS(R4)
          0A 1C A4   1A   E1  1386          BBC     #CLUB$V_QF_NEWVOTE, -      ; Branch if no quorum disk membership
                              138B                  CLUB$L_FLAGS(R4),10$
          22 A4   46 A4   A0  138B          ADDW    CLUB$W_NEWQDVOTES(R4), -   ; Count quorum disk in votes
                              1390                  CLUB$W_VOTES(R4)
          00 1C A4   19   E2  1390          BBSS    #CLUB$V_QF_VOTE, -         ; Mark quorum disk a member
                              1395                  CLUB$L_FLAGS(R4),10$
                EC68'   30  1395  10$:    BSBW    CNX$SCAN_CSBS              ; Iterate over all CSBs
```

```
            0E 50    E9  1398  4425          BLBC    R0,30$                    ; Branch when done
   08 60 A3    11    E1  139B  4426          BBC     #CSB$V_SELECTED, -        ; Branch if not selected
                         13A0  4427                  CSB$L_STATUS(R3),20$
            24 A4    B6  13A0  4428          INCW    CLUB$W_NODES(R4)          ; Count this node
   22 A4    50 A3    A0  13A3  4429          ADDW    CSB$W_VOTES(R3), -        ; Count votes for this node
                         13A8  4430                  CLUB$W_VOTES(R4)
                     05  13A8  4431 20$:     RSB
                         13A9  4432
                0C   BA  13A9  4433 30$:     POPR    #^M<R2,R3>                ; Restore registers
                     05  13AB  4434          RSB                               ; Return
```

```
                         13AC          .SBTTL  ADD_NODE - Make node a cluster member
                         13AC  ;++
                         13AC  ;
                         13AC  ; FUNCTIONAL DESCRIPTION:
                         13AC  ;
                         13AC  ;       Make the specified node a cluster member.
                         13AC  ;
                         13AC  ; CALLING SEQUENCE:
                         13AC  ;
                         13AC  ;       JSB     ADD_NODE
                         13AC  ;       IPL is IPL$_SCS
                         13AC  ;
                         13AC  ; INPUT PARAMETERS:
                         13AC  ;
                         13AC  ;       R3:     CSB of the node to be added
                         13AC  ;               CSB$L_CSID contains CSID for the node
                         13AC  ;
                         13AC  ; OUTPUT PARAMETERS:
                         13AC  ;
                         13AC  ;       R4:     Address of CLUB
                         13AC  ;
                         13AC  ; COMPLETION CODES:
                         13AC  ;
                         13AC  ;       NONE
                         13AC  ;
                         13AC  ; SIDE EFFECTS:
                         13AC  ;
                         13AC  ;       The cluster vector is updated.
                         13AC  ;       The CSB reference count is incremented.
                         13AC  ;       R0-R1 are destroyed.
                         13AC  ;
                         13AC  ;--
                         13AC
                         13AC  ADD_NODE:
         6C A3    96     13AC          INCB    CSB$B_REF_CNT(R3)         ; Bump reference count to nail down CSB
      60 A3    02    C8  13AF          BISL2   #CSB$M_MEMBER, -          ; Mark the node a cluster member
                         13B3                  CSB$L_STATUS(R3)
      50    4C A3    3C  13B3          MOVZWL  CSB$W_CSID_IDX(R3),R0     ; CSID slot index
               3C    13  13B7          BEQL    90$                       ; Branch if 0 -- invalid index
 00000000'GF   50    B1  13B9          CMPW    R0,G^CLU$GW_MAXINDEX      ; Valid index
               33    1E  13C0          BGEQU   90$                       ; Branch if invalid index
 51   00000000'GF    D0  13C2          MOVL    G^CLU$GL_CLUSVEC,R1       ; Address of cluster vector
             6140    D5  13C9          TSTL    (R1)[R0]                  ; Is slot free?
               27    19  13CC          BLSS    90$                       ; Branch if slot in use
       6140    53    D0  13CE          MOVL    R3,(R1)[R0]               ; Store CSB address in slot
        54    64 A3  D0  13D2          MOVL    CSB$L_CLUB(R3),R4         ; Address of CLUB
   05 60 A3    08    E0  13D6          BBS     #CSB$V_CLUSTER, -         ; Skip ref time update if remote
                         13DB                  CSB$L_STATUS(R3),5$       ;   node is already a cluster member
   74 A3    3C A4    7D  13DB          MOVQ    CLUB$Q_LST_TIME(R4), -    ; Update reference time
                         13E0                  CSB$Q_REFTIME(R3)
   05 60 A3    18    E1  13E0  5$:     BBC     #CSB$V_LOCAL, -           ; Branch if not the local node
                         13E5                  CSB$L_STATUS(R3),10$
   60 A4    4C A3    D0  13E5          MOVL    CSB$L_CSID(R3), -         ; Store local CSID in CLUB
                         13EA                  CLUB$L_LOCAL_CSID(R4)
   05 60 A3    00    E1  13EA  10$:    BBC     #CSB$V_LONG_BREAK, -      ; Branch if no long break seen
                         13EF                  CSB$L_STATUS(R3),20$
   00 1C A4    17    E2  13EF          BBSS    #CLUB$V_LOST_CNX, -       ; Mark cluster connection lost
```

```
        13F4   4493                         CLUB$L_FLAGS(R4),20$
  05    13F4   4494 20$:     RSB
        13F5   4495
        13F5   4496 90$:     BUG_CHECK      CNXMGRERR,FATAL ; Consistency check
```

```
                                  13F9        .SBTTL  REMOVE_NODE - Remove a node from the cluster
                                  13F9  ;++
                                  13F9  ;
                                  13F9  ; FUNCTIONAL DESCRIPTION:
                                  13F9  ;
                                  13F9  ;       Remove the specified node from the cluster
                                  13F9  ;
                                  13F9  ; CALLING SEQUENCE:
                                  13F9  ;
                                  13F9  ;       JSB     REMOVE_NODE
                                  13F9  ;       IPL is IPL$_SCS
                                  13F9  ;
                                  13F9  ; INPUT PARAMETERS:
                                  13F9  ;
                                  13F9  ;       R3:     CSB of the node to be removed
                                  13F9  ;
                                  13F9  ; OUTPUT PARAMETERS:
                                  13F9  ;
                                  13F9  ;       R3:     If CSB is deleted, the contents of back pointer cell.
                                  13F9  ;               If CSB is not deleted, R3 is preserved.
                                  13F9  ;
                                  13F9  ; COMPLETION CODES:
                                  13F9  ;
                                  13F9  ;       NONE
                                  13F9  ;
                                  13F9  ; SIDE EFFECTS:
                                  13F9  ;
                                  13F9  ;       The cluster vector is updated.
                                  13F9  ;       The CSB reference count is decremented.
                                  13F9  ;       R0-R2 are destroyed.
                                  13F9  ;
                                  13F9  ;--
                                  13F9
                                  13F9  REMOVE_NODE:
                       30    BB   13F9          PUSHR   #^M<R4,R5>              ; Save register
                    55 53    D0   13FB          MOVL    R3,R5                   ; Address of CSB
           50   0000'CF    9E   13FE          MOVAB   W^FAILOVER_MSG,R0       ; Message for removing node from cluster
                 EBFA'    30   1403          BSBW    CNX$CONFIG_CHANGE       ; Tell the world the node is gone
        50 60 A3    01    E5   1406          BBCC    #CSB$V_MEMBER, -        ; Mark the node not a cluster member
                                  140B                  CSB$L_STATUS(R3), 90$   ; *** Is this right?
           60 A3    04    C8   140B          BISL2   #CSB$M_REMOVED, -       ; Mark the node removed
                                  140F                  CSB$L_STATUS(R3)
     74 A3    3C A4    7D   140F          MOVQ    CLUB$Q_LST_TIME(R4), -  ; Update reference time
                                  1414                  CSB$Q_REFTIME(R3)
          54    64 A3    D0   1414          MOVL    CSB$L_CLUB(R3),R4       ; Fetch address of CLUB
                                  1418  ;
                                  1418  ; Tell quorum disk manager to skip a read cycle since a node has just been dropped
                                  1418  ; from the cluster -- the removed node may still have a chance to write.
                                  1418  ;
  1C A4   01000000 8F    C8   1418          BISL2   #CLUB$M_QF_FAILED_NODE, - ; Set bit requesting that a read cycle be
                                  1420                  CLUB$L_FLAGS(R4)        ;   skipped
  1C A4   40000000 8F    CA   1420          BICL2   #CLUB$M_QF_DYNVOTE, -   ; Stop dynamically counting the quorum disk
                                  1428                  CLUB$L_FLAGS(R4)        ;   until remote node can sense us
          50    4C A3    3C   1428          MOVZWL  CSB$W_CSID_IDX(R3),R0   ; CSID slot index
                       2D    13   142C          BEQL    90$                     ; Branch if 0 -- invalid index
     00000000'GF    50    B1   142E          CMPW    R0,G^CLU$GW_MAXINDEX    ; Valid index
                       24    1E   1435          BGEQU   90$                     ; Branch if invalid index
```

```
 51   00000000'GF   D0  1437          MOVL    G^CLU$GL_CLUSVEC,R1         ; Address of cluster vector
        53   6140   D1  143E          CMPL    (R1)[R0],R3                 ; Is slot consistent with CSB?
               17   12  1442          BNEQ    90$                         ; Branch if things are inconsistent
   6140   4E   A3   3C  1444          MOVZWL  CSB$W_CSID_SEQ(R3), -       ; Store old sequence number for reuse
                        1449                  (R1)[R0]
          55   53   D0  1449          MOVL    R3,R5                       ; Address of CSB
               EBB1'  30  144C        BSBW    CNX$DISC_REMOVE             ; Break connection to node that has been rem
               EBAE'  30  144F        BSBW    SEND_JBCMSG                 ; Tell the job controller that the node is g
               EBAB'  30  1452        BSBW    CNX$DECREFCNT               ; Decrement reference count, perhaps delete
          53   55   D0  1455          MOVL    R5,R3                       ; Address of this/previous CSB
               30   BA  1458          POPR    #^M<R4,R5>                  ; Restore R4 and R5
                    05  145A          RSB
                        145B
                        145B     90$: BUG_CHECK       CNXMGRERR,FATAL     ; Consistency check
                        145F
                        145F
                        145F          .END
```

```
$$BASE                          = 00000001
$$DISPL                         = 00000005
$$GENSW                         = 00000001
$$HIGH                          = 00000004
$$LIMIT                         = 00000003
$$LOW                           = 00000001
$$MNSW                          = 00000001
$$MXSW                          = 00000001
ADDNODE_MSG                       ********   X    03
ADD_NODE                          000013AC R      03
ADJUST_QUORUM                     00000B19 R      03
ANALYZE_PHASE                     00000137 R      03
BLD_DESC_MSG                      00000F87 R      03
BLD_FORM_MSG                      00000BFA R      03
BLD_JOIN_MSG                      00000BFA R      03
BLD_LOCK_MSG                      00001198 R      03
BLD_QUORUM_MSG                    00000BFA R      03
BLD_RECONFIG_MSG                  00000BFA R      03
BLD_STD_MSG                       00001339 R      03
BLD_TOPOLOGY_MSG                  00001022 R      03
BLD_VEC_MSG                       00000F04 R      03
BUG$_CLUEXIT                      ********   X    03
BUG$_CNXMGRERR                    ********   X    03
BUG$_OPERATOR                     ********   X    03
BUILD_STS_MSG                     000002C9 R      03
CDRP$L_MSGBLD                   = 0000004C
CDRP$L_RSPID                    = 00000020
CDRP$L_VAL1                     = 0000002C
CDRP$L_VAL2                     = 00000030
CDRP$L_VAL3                     = 00000034
CDRP$L_VAL4                     = 00000038
CDRP$L_VAL5                     = 0000003C
CDRP$L_VAL7                     = 00000044
CDRP$L_VAL8                     = 00000048
CLMCNX$B_ACK                    = 00000012
CLMCNX$B_REPLY                  = 00000013
CLMCNX$B_XTN_CODE               = 00000011
CLMCNX$B_XTN_PHASE              = 00000010
CLMCNX$K_DATA                   = 00000030
CLMCNX$K_FNC_DESC               = 00000005
CLMCNX$K_FNC_ENTER              = 00000002
CLMCNX$K_FNC_FORM               = 00000007
CLMCNX$K_FNC_JOIN               = 00000009
CLMCNX$K_FNC_LOCK               = 00000003
CLMCNX$K_FNC_PH2                = 0000000A
CLMCNX$K_FNC_QUORUM             = 0000000D
CLMCNX$K_FNC_RECONFIG           = 00000008
CLMCNX$K_FNC_STATUS             = 00000001
CLMCNX$K_FNC_TOPOLOGY           = 0000000F
CLMCNX$K_FNC_TRNSTS             = 0000000E
CLMCNX$K_FNC_UNLOCK             = 00000004
CLMCNX$K_FNC_VEC                = 00000006
CLMCNX$K_IDLE                   = 00000010
CLMCNX$K_LOCK                   = 00000020
CLMCNX$K_PH1                    = 00000040
CLMCNX$K_PH2                    = 00000060
CLMCNX$K_RP_TRNSTS_PH0          = 00000001
CLMCNX$K_RP_TRNSTS_PH1          = 00000003
CLMCNX$K_RP_TRNSTS_PH1B         = 00000002
CLMCNX$K_RP_TRNSTS_PH2          = 00000004
CLMCNX$K_UNLOCK                 = 00000050
CLMCNX$K_XTN_FORM               = 00000001
CLMCNX$K_XTN_JOIN               = 00000002
CLMCNX$K_XTN_QUORUM             = 00000004
CLMCNX$K_XTN_RECONFIG           = 00000003
CLMCNX$L_XTN_ID                 = 0000000C
CLMCNX$V_RP_TRNSTS_CMT          = 00000000
CLMLCK$Q_XTN_TIME               = 00000014
CLMNOD$B_SYSTEMID               = 00000014
CLMNOD$L_CSID                   = 00000024
CLMNOD$Q_SWINCARN               = 0000001C
CLMNOD$S_SYSTEMID               = 00000006
CLMPRO$B_FLAGS                  = 00000030
CLMPRO$B_FSYSID                 = 00000031
CLMPRO$B_NODEMAP                = 00000037
CLMPRO$L_FMERIT                 = 0000001C
CLMPRO$M_QF_VOTE                = 00000001
CLMPRO$Q_CURTIME                = 00000028
CLMPRO$Q_FTIME                  = 00000020
CLMPRO$S_FSYSID                 = 00000006
CLMPRO$S_NODEMAP                = 00000020
CLMPRO$V_QF_VOTE                = 00000000
CLMPRO$W_MEMSEQ                 = 00000018
CLMPRO$W_NEXT_CSID              = 00000014
CLMPRO$W_QDVOTES                = 0000001A
CLMPRO$W_QUORUM                 = 00000016
CLMSTS$B_FLAGS                  = 0000000C
CLMSTS$B_QDISK                  = 00000030
CLMSTS$L_MAX_XTN                = 0000002C
CLMSTS$M_CLUSTER                = 00000001
CLMSTS$M_QF_ACTIVE              = 00000002
CLMSTS$M_SHUTDOWN               = 00000004
CLMSTS$Q_FTIME                  = 0000001C
CLMSTS$Q_LST_TIME               = 00000024
CLMSTS$Q_REFTIME                = 00000040
CLMSTS$S_QDISK                  = 00000010
CLMSTS$V_CLUSTER                = 00000000
CLMSTS$V_QF_ACTIVE              = 00000001
CLMSTS$V_SHUTDOWN               = 00000002
CLMSTS$W_CQUORUM                = 0000000E
CLMSTS$W_CVOTES                 = 00000010
CLMSTS$W_LCKDIRWT               = 0000001A
CLMSTS$W_NODES                  = 00000012
CLMSTS$W_NQUORUM                = 00000014
CLMSTS$W_NVOTES                 = 00000016
CLMSTS$W_QDVOTES                = 00000018
CLMTOP$B_NODEMAP                = 00000014
CLMTOP$S_NODEMAP                = 00000020
CLMVEC$W_INDEX                  = 00000014
CLMVEC$W_SEQUENCE               = 00000016
CLSMSG$B_FACILITY               = 00000008
CLSMSG$B_FUNC                   = 00000009
CLSMSG$K_FAC_CNX                = 00000001
CLSMSG$M_RESPMSG                = 00000080
```

```
CLU$GL_CLUB                  ********   X  04
CLU$GL_CLUSVEC               ********   X  03
CLU$GW_MAXINDEX              ********   X  03
CLUB$B_CLUBPWF             = 0000018C
CLUB$B_CLUFCB              = 0000010C
CLUB$B_CUR_CODE            = 00000058
CLUB$B_CUR_PHASE           = 00000059
CLUB$B_FORK_BLOCK          = 000000CC
CLUB$B_FSYSID              = 00000026
CLUB$B_LST_CODE            = 00000044
CLUB$B_LST_PHASE           = 00000045
CLUB$B_NODEMAP             = 000000EC
CLUB$L_COORD               = 0000005C
CLUB$L_CSBQFL              = 00000000
CLUB$L_CTX0                = 00000070
CLUB$L_CTX1                = 00000078
CLUB$L_CUR_COORD           = 0000004C
CLUB$L_CUR_XTN             = 00000048
CLUB$L_FLAGS               = 0000001C
CLUB$L_FMERIT              = 000000A8
CLUB$L_FOREIGN_CLUSTER     = 000000C8
CLUB$L_LOCAL_CSB           = 00000010
CLUB$L_LOCAL_CSID          = 00000060
CLUB$L_LST_COORD           = 00000038
CLUB$L_LST_XTN             = 00000034
CLUB$L_MAX_XTN             = 00000068
CLUB$L_RETT                = 00000074
CLUB$L_TQE                 = 00000084
CLUB$M_BACKOUT             = 00100000
CLUB$M_LOST_CNX            = 00800000
CLUB$M_QF_DYNVOTE          = 40000000
CLUB$M_QF_FAILED_NODE      = 01000000
CLUB$M_QF_NEWVOTE          = 04000000
CLUB$M_QF_VOTE             = 02000000
CLUB$M_QUORUM              = 10000000
CLUB$M_SHUTDOWN            = 00000004
CLUB$M_STS_PH0             = 00000200
CLUB$M_STS_PH1             = 00000800
CLUB$M_STS_PH1B            = 00000400
CLUB$M_STS_PH2             = 00001000
CLUB$M_STS_PPHASE          = 00000100
CLUB$M_TRANSITION          = 20000000
CLUB$M_UNLOCK              = 00020000
CLUB$Q_CUR_TIME            = 00000050
CLUB$Q_FTIME               = 0000002C
CLUB$Q_LST_TIME            = 0000003C
CLUB$Q_NEWTIME             = 00000094
CLUB$Q_NEWTIME_REF         = 0000009C
CLUB$S_FSYSID              = 00000006
CLUB$S_NODEMAP             = 00000020
CLUB$T_QDNAME              = 000000B8
CLUB$V_ADJ_QUORUM          = 0000001B
CLUB$V_BACKOUT             = 00000014
CLUB$V_CLUSTER             = 00000000
CLUB$V_INIT                = 00000013
CLUB$V_LOST_CNX            = 00000017
CLUB$V_NO_FORM             = 00000012

CLUB$V_QF_ACTIVE           = 00000001
CLUB$V_QF_DYNVOTE          = 0000001E
CLUB$V_QF_FAILED_NODE      = 00000018
CLUB$V_QF_NEWVOTE          = 0000001A
CLUB$V_QF_VOTE             = 00000019
CLUB$V_QUORUM              = 0000001C
CLUB$V_SHUTDOWN            = 00000002
CLUB$V_STS_PH0             = 00000009
CLUB$V_STS_PH1             = 0000000B
CLUB$V_STS_PH1B            = 0000000A
CLUB$V_STS_PH2             = 0000000C
CLUB$V_STS_PPHASE          = 00000008
CLUB$V_TRANSITION          = 0000001D
CLUB$V_UNLOCK              = 00000011
CLUB$W_ADJ_QUORUM          = 000000A6
CLUB$W_FIRST_INDEX         = 00000066
CLUB$W_MEMSEQ              = 000000AC
CLUB$W_MSGCNT              = 0000005A
CLUB$W_NEWQDVOTES          = 00000046
CLUB$W_NEWQUORUM           = 000000A4
CLUB$W_NEXT_CSID           = 00000064
CLUB$W_NODES               = 00000024
CLUB$W_QDVOTES             = 000000AE
CLUB$W_QUORUM              = 00000020
CLUB$W_VOTES               = 00000022
CLUBFKB$B_FORK_BLOCK       = 00000000
CLUBFKB$L_STATUS           = 0000001C
CLUBFKB$V_FKB_BUSY         = 00000000
CLUBPWF$L_STATUS           = 00000018
CLUBPWF$V_BUSY             = 00000000
CLUFCB$B_NODEMAP           = 00000028
CNX$ALLOC_CDRP_ONLY          ********   X  03
CNX$ALLOC_WARMCDRP_CSB       ********   X  03
CNX$ALLOZMEM                 ********   X  04
CNX$ASSIGN_CSID              ********   X  03
CNX$BUGCHECK_CLUSTER         ********   X  03
CNX$CHANGE_QUORUM            00000226 RG   03
CNX$CHECK_QUORUM             ********   X  03
CNX$CHECK_UNLOCK             00001234 RG   03
CNX$CLUB_FORK                ********   X  03
CNX$CLUB_WAIT                ********   X  03
CNX$CONFIG_CHANGE            ********   X  03
CNX$CON_BREAK                000000C5 RG   03
CNX$CON_INIT                 00000000 RG   04
CNX$CON_NEWSYS               000000B5 RG   03
CNX$DEALL_MSG_BUF_CSB        ********   X  03
CNX$DEALL_WARMCDRP_CSB       ********   X  03
CNX$DECREFCNT                ********   X  03
CNX$DIRVEC_ADJ               ********   X  03
CNX$DIRVEC_FILL              ********   X  03
CNX$DISC_REMOVE              ********   X  03
CNX$DISK_CHANGE              000001EF RG   03
CNX$FIX_EPID                 ********   X  03
CNX$INIT_CDRP                ********   X  03
CNX$INIT_CSBS                ********   X  03
CNX$INIT_STD_RESP            00001309 RG   03
CNX$MARK_LOCKED              ********   X  03
```

```
CNX$MARK_UNLOCKED              ********    X    03
CNX$MEMBERSHIP_CHANGE          ********    X    03
CNX$OPT                        ********    X    03
CNX$OPT_INIT                   ********    X    03
CNX$POWER_FAIL                 0000006E RG      03
CNX$PROCESS_RESPONSE           000012E6 RG      03
CNX$QUORUM_CALC                ********    X    03
CNX$RANDOM                     ********    X    03
CNX$RANDOM_INIT                ********    X    03
CNX$RCVD_DESC                  00000FA7 RG      03
CNX$RCVD_ENTER                 000008D9 RG      03
CNX$RCVD_FORM                  00000C4F RG      03
CNX$RCVD_JOIN                  00000DD8 RG      03
CNX$RCVD_LOCK                  000011A5 RG      03
CNX$RCVD_PH2                   000010F1 RG      03
CNX$RCVD_QUORUM                00000EAA RG      03
CNX$RCVD_RECONFIG              00000D0C RG      03
CNX$RCVD_STATUS                0000033A RG      03
CNX$RCVD_TOPOLOGY              00001039 RG      03
CNX$RCVD_TRNSTS                0000044A RG      03
CNX$RCVD_UNLOCK                000012BC RG      03
CNX$RCVD_VEC                   00000F27 RG      03
CNX$RESOURCE_CHECK             ********    X    03
CNX$RESP_FORGET                ********    X    03
CNX$SCAN_CSBS                  ********    X    03
CNX$SCAN_CSBS_EXIT             ********    X    03
CNX$SCAN_CSBS_FORK             ********    X    03
CNX$SCAN_CSBS_RETRY            ********    X    03
CNX$SEND_ALL_STATUS            00000275 RG      03
CNX$SEND_FORGET                ********    X    03
CNX$SEND_MSG_CSB               ********    X    03
CNX$SHUTDOWN                   00000249 RG      03
COMPLETE_MSG                   ********    X    03
CSB$B_NODEMAP                = 0000008C
CSB$B_REF_CNT                = 0000006C
CSB$L_CLUB                   = 00000064
CSB$L_CSID                   = 0000004C
CSB$L_SB                     = 00000068
CSB$L_STATUS                 = 00000060
CSB$L_SYSQFL                 = 00000000
CSB$M_CLUSTER                = 00000100
CSB$M_LONG_BREAK             = 00000001
CSB$M_MEMBER                 = 00000002
CSB$M_QF_ACTIVE              = 00000200
CSB$M_QF_SAME                = 00000008
CSB$M_REMOVED                = 00000004
CSB$M_SELECTED               = 00020000
CSB$M_SEND_STATUS            = 04000000
CSB$M_SHUTDOWN               = 00000400
CSB$M_STATUS_RCVD            = 02000000
CSB$Q_REFTIME                = 00000074
CSB$S_NODEMAP                = 00000020
CSB$V_CLUSTER                = 00000008
CSB$V_LOCAL                  = 00000018
CSB$V_LOCKED                 = 00000010
CSB$V_LONG_BREAK             = 00000000
CSB$V_MEMBER                 = 00000001
CSB$V_QF_ACTIVE              = 00000009
CSB$V_QF_SAME                = 00000003
CSB$V_SELECTED               = 00000011
CSB$V_SEND_STATUS            = 0000001A
CSB$V_SHUTDOWN               = 0000000A
CSB$V_STATUS_RCVD            = 00000019
CSB$W_CSID_IDX               = 0000004C
CSB$W_CSID_SEQ               = 0000004E
CSB$W_LCKDIRWT               = 00000054
CSB$W_NODES                  = 0000006E
CSB$W_QDVOTES                = 00000056
CSB$W_QUORUM                 = 00000052
CSB$W_VOTES                  = 00000050
DESCRIBE_NODE                  00000F4B R       03
DO_PHASE2                      00001108 R       03
DO_TIMEOUT                     000004BB R       03
DYN$C_FRK                    = 00000008
DYN$C_TQE                    = 0000000F
EXE$ALONONPAGED                ********    X    03
EXE$AL_TQENOREPT               ********    X    03
EXE$CLOTRANIO                  ********    X    03
EXE$DEANONPAGED                ********    X    03
EXE$FORK                       ********    X    03
EXE$FORK_WAIT                  ********    X    03
EXE$GQ_SYSTIME                 ********    X    03
EXE$INSTIMQ                    ********    X    03
FAILOVER_MSG                   ********    X    03
FKB$B_FIPL                   = 0000000B
FKB$B_TYPE                   = 0000000A
FKB$K_LENGTH                 = 00000018
FORM_CLUSTER                   0000060F R       03
FORM_DESCRIBE                  00000687 R       03
FORM_FINISH                    00000761 R       03
FORM_JOIN_FINISH               00000A86 R       03
FORM_PROPOSE                   0000070F R       03
INIT_CONTINUE                  00000000 R       03
INIT_STD_MSG                   00001312 R       03
INIT_TRANSITION                00000BA1 R       03
IPL$_SCS                     = 00000008
IPL$_SYNCH                   = 00000008
IPL$_TIMER                   = 00000008
JOIN_CLUSTER                   00000887 R       03
JOIN_FINISH                    00000A86 R       03
JOIN_LOCKED                    00000974 R       03
JOIN_MSG                       ********    X    03
JOIN_PROPOSE                   00000A4F R       03
LCK$GB_STALLREQS               ********    X    03
LOCK_NODES                     00001137 R       03
MEMREQ_MSG                     ********    X    03
MSG_CHECK                      00001349 R       03
PDT$L_POLLSWEEP              = 000000D8
PR$_IPL                        ********    X    03
QUORUM_FINISH                  00000B80 R       03
QUORUM_MSG                     ********    X    03
RCVSTS_MSG                     ********    X    03
RECONFIG_CLUSTER               00000764 R       03
RECONFIG_FINISH                00000825 R       03
```

```
RECONFIG_LOCKED                  0000078E R     03
RECONFIG_MSG                     ********   X   03
RECONFIG_PROPOSE                 000007E8 R     03
REMOVE_NODE                      000013F9 R     03
REQJOIN_MSG                      ********   X   03
SB$B_SYSTEMID                  = 00000018
SB$Q_SWINCARN                  = 0000002C
SB$S_SWINCARN                  = 00000008
SB$S_SYSTEMID                  = 00000006
SCAN                             00000555 R     03
SCS$DEALL_RSPID                  ********   X   03
SCS$GA_LOCALSB                   ********   X   03
SCS$GL_PDT                       ********   X   03
SCS$GW_PRCPOLINT                 ********   X   03
SEND_JBCMSG                      ********   X   03
SEND_PH1                         0000105F R     03
SEND_PH2                         000010BE R     03
SEND_STATUS                      00000299 R     03
SNDSTS_MSG                       ********   X   03
SS$_NORMAL                     = 00000001
START_LONG_TIMEOUT               000004AF R     03
START_TIMEOUT                    000004B6 R     03
TQE$B_RQTYPE                   = 0000000B
TQE$B_TYPE                     = 0000000A
TQE$C_SSSNGL                   = 00000001
TQE$K_LENGTH                   = 00000030
TQE$L_FPC                      = 0000000C
TQE$L_FR4                      = 00000014
TQE$W_SIZE                     = 00000008
TRYFORM_MSG                      ********   X   03
UNLOCK_ALL                       00001246 R     03
UNLOCK_MSG                       ********   X   03
UNLOCK_NODE                      000012C2 R     03
UPDATE_QUORUM                    00001363 R     03

                                                  +----------------+
                                                  ! 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
$$$040                          00000000  (    0.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC LONG
$$$100                          0000145F  ( 5215.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC LONG
$$$002                          00000037  (   55.)  04 (  4.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC LONG

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

Phase                     Page faults   CPU Time       Elapsed Time
-----                     -----------   --------       ------------
Initialization                     30   00:00:00.06    00:00:00.80
Command processing                119   00:00:00.43    00:00:02.85
Pass 1                            633   00:00:17.20    00:01:08.97
Symbol table sort                   8   00:00:02.20    00:00:07.76
```

```
Pass 2                         529      00:00:06.64     00:00:26.34
Symbol table output              3      00:00:00.21     00:00:01.19
Psect synopsis output            0      00:00:00.02     00:00:00.02
Cross-reference output           0      00:00:00.00     00:00:00.00
Assembler run totals          1324      00:00:26.76     00:01:47.93

The working set limit was 1950 pages.
154996 bytes (303 pages) of virtual memory were used to buffer the intermediate code.
There were 110 pages of symbol table space allocated to hold 1826 non-local and 258 local symbols.
4571 source lines were read in Pass 1, producing 40 object records in Pass 2.
32 pages of virtual memory were used to define 30 macros.

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

Macro library name                      Macros defined
------------------                      --------------
_$255$DUA28:[SYSLOA.OBJ]CLUSTER.MLB;1              2
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                    17
_$255$DUA28:[SYSLIB]STARLET.MLB;2                  5
TOTALS (all libraries)                            24

1876 GETS were required to define 24 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:CONMAN/OBJ=OBJ$:CONMAN MSRC$:CONMAN/UPDATE=(ENH$:CONMAN)+EXECML$/LIB+LIB$:CLUSTER/LIB
```

0392 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CNXOPT
LIS

ADPSUBUV1
LIS

CNXMAN
LIS

CJFCLUSTR
LIS

CLUSTRLOA
LIS

CONMAN
LIS

ADPSUB790
LIS

CLUMESSAG
LIS

0393 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CSPCALL
LIS

CSPCJFMAS
LIS

CSP
LIS

CSPBRKTHR
LIS

CONUTIL
LIS

CSPCALLACT
LIS

CONSUBS
LIS