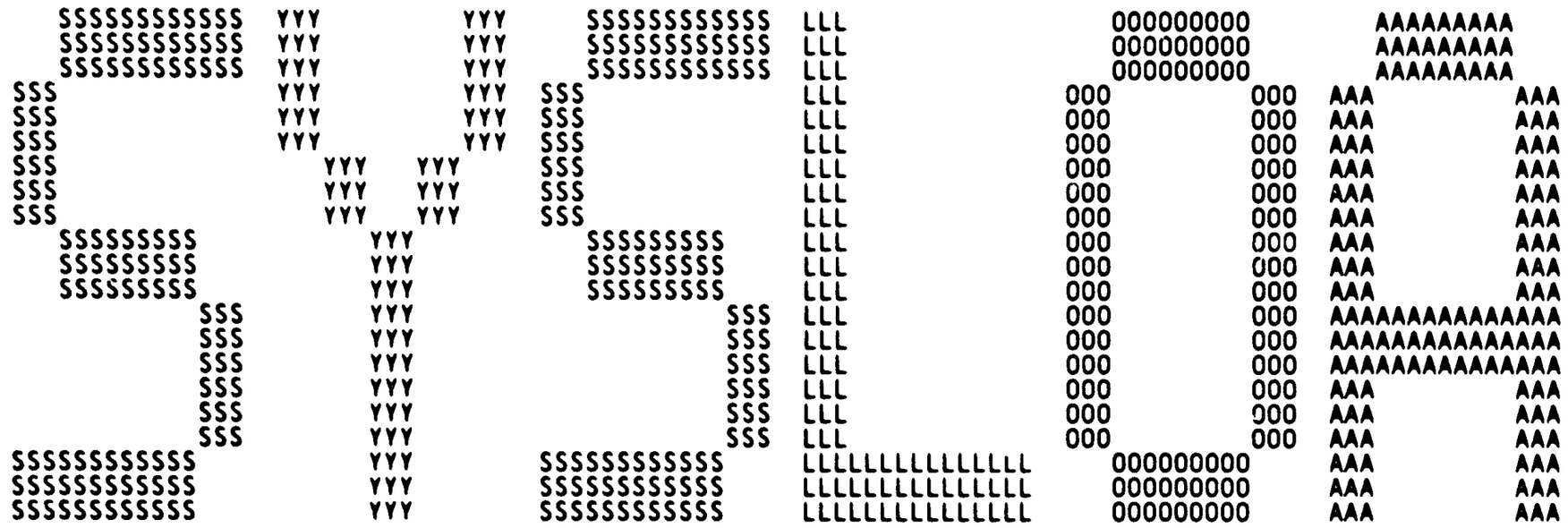
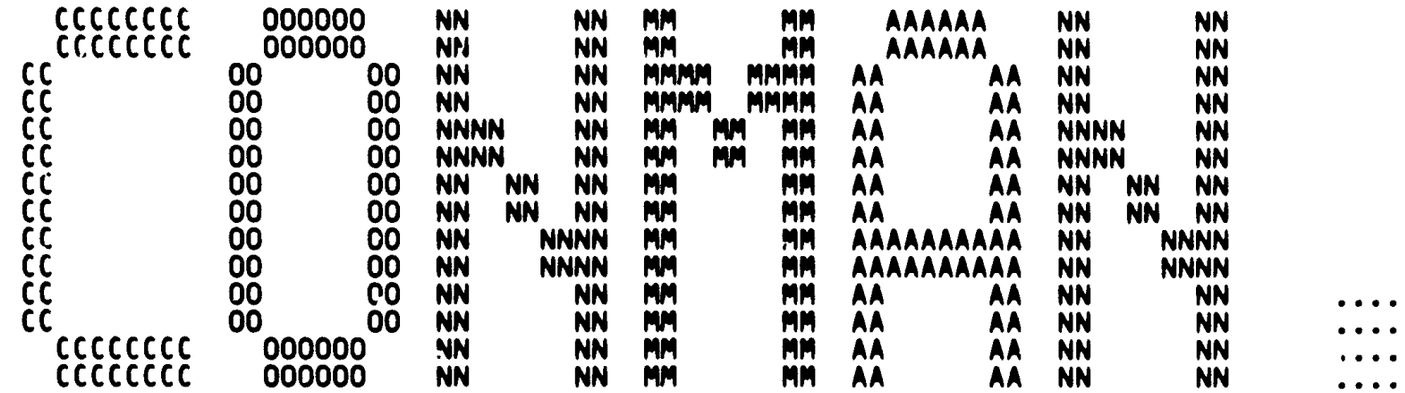

SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL 000000000 AAAAAAAAA
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL 000000000 AAAAAAAAA
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL 000000000 AAAAAAAAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSS YYY YYY SSS LLL 000 000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


CCCCCCCC 000000 NN NN MM MM AAAAAA NN NN
CCCCCCCC 000000 NN NN MM MM AAAAAA NN NN
CC 00 00 NN NN MMMM MMMM AA AA NN NN
CC 00 00 NN NN MMMM MMMM AA AA NN NN
CC 00 00 NNNN NN MM MM MM AA AA NNNN NN
CC 00 00 NNNN NN MM MM MM AA AA NNNN NN
CC 00 00 NN NN NN MM MM AA AA NN NN NN
CC 00 00 NN NN NN MM MM AA AA NN NN NN
CC 00 00 NN NNNN MM MM AAAAAAAAAA NN NNNN
CC 00 00 NN NNNN MM MM AAAAAAAAAA NN NNNN
CC 00 00 NN NN MM MM AA AA NN NN
CC 00 00 NN NN MM MM AA AA NN NN
CCCCCCCC 000000 NN NN MM MM AA AA NN NN
CCCCCCCC 000000 NN NN MM MM AA AA NN NN

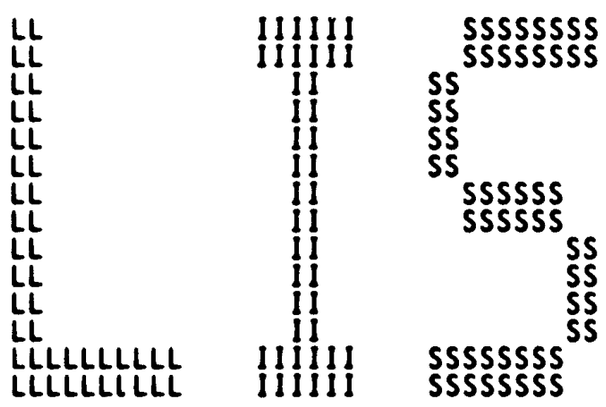

LL IIIIII SSSSSSS
LL IIIIII SSSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LL II SSSSS
LL II SSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LLLLLLLLLL IIIIII SSSSSSS
LLLLLLLLLL IIIIII SSSSSSS

Table of contents

(2) 187 DECLARATIONS
(3) 225 CNX$CON_INIT - Initialization Call
(4) 343 CNX$POWER_FAIL - Power Fail Recovery Entry
(5) 408 CNX$CON_NEWSYS - Connected to New System
(6) 452 CNX$CON_BREAK - Connection has broken
(7) 555 ANALYZE_PHASE - Determine Transition Phase and Finish Transition
(8) 721 CNX$DISK_CHANGE - Quorum Disk Connection State Change
(9) 775 CNX$CHANGE_QUORUM - Adjust quorum
(10) 830 CNX$SHUTDOWN - Request cluster shutdown
(11) 884 CNX$SEND_ALL_STATUS - Send Status to All Flagged Systems
(12) 940 SEND_STATUS - Send Status to a System
(13) 998 BLD_STS_MSG - Build Status Message
(14) 1082 CNX$RCVD_STATUS - Status Received from a System
(15) 1261 CNX$RCVD_TRNSTS - Received Transition Status Request Message
(16) 1353 START_TIMEOUT - Start random CLUB-based timeout
(16) 1354 START_LONG_TIMEOUT - Start long random CLUB-based timeout
(17) 1452 SCAN - Look for work to do
(18) 1581 FORM_CLUSTER - Try to form a cluster
(19) 1789 RECONFIG_CLUSTER - Reconfigure a cluster following a node failure
(20) 1995 JOIN_CLUSTER - Try to join a cluster
(21) 2061 CNX$RCVD_ENTER - Received a cluster membership request
(22) 2370 ADJUST_QUORUM - Adjust Cluster Quorum and/or Quorum Disk Membership
(23) 2487 INIT_TRANSITION - Initialization for a transition
(24) 2551 BLD_FORM_MSG - Build Message Proposing a New Cluster
(24) 2552 BLD_RECONFIG_MSG - Build Message Proposing a Reconfigured Cluster
(24) 2553 BLD_JOIN_MSG - Build Message Proposing a New Node in a Cluster
(24) 2554 BLD_QUORUM_MSG - Build Message Proposing a Quorum/Quorum Disk Chanage
(25) 2635 CNX$RCVD_FORM - Cluster formation proposal received
(26) 2747 CNX$RCVD_RECONFIG - Cluster Reconfiguration proposal received
(27) 2862 CNX$RCVD_JOIN - Cluster node addition proposal received
(28) 2981 CNX$RCVD_QUORUM - Quorum Update Message Received
(29) 3052 BLD_VEC_MSG - Build Message Describing Vector Slot
(30) 3108 CNX$RCVD_VEC - Cluster vector slot information received
(31) 3155 DESCRIBE_NODE - Describe one node to another node
(32) 3223 BLD_DESC_MSG - Build Message describing a node
(33) 3281 CNX$RCVD_DESC - Node description received
(34) 3376 BLD_TOPOLOGY_MSG - Build Message Describing Cluster Topology
(35) 3432 CNX$RCVD_TOPOLOGY - Topology Request Message Received
(36) 3484 SEND_PH1 - Send Phase 1 Messages
(37) 3576 SEND_PH2 - Send a Phase 2 Notification
(38) 3633 CNX$RCVD_PH2 - Phase 2 request received
(39) 3707 LOCK_NODES - Obtain Coordinator Lock on Selected Nodes
(40) 3799 BLD_LOCK_MSG - Build Coordinator Lock Request Message
(41) 3850 CNX$RCVD_LOCK - Lock request received
(42) 3947 CNX$CHECK_UNLOCK - Check UNLOCK flag
(43) 3989 UNLOCK_ALL - Release Coordinator Lock on All Nodes
(44) 4074 CNX$RCVD_UNLOCK - Unlock request received
(45) 4131 CNX$PROCESS_RESPONSE - Process simple response message
(46) 4187 INIT_STD_MSG - Common CDRP initialization for standard message
(46) 4188 CNX$INIT_STD_RESP - Common CDRP initialization for standard response
(47) 4260 BLD_STD_MSG - Build Standard Message from CDRP
(48) 4320 MSG_CHECK - Standard incoming message verification
(49) 4368 UPDATE_QUORUM - Update Quorum Parameters in CLUB
(50) 4436 ADD_NODE - Make node a cluster member
(51) 4498 REMOVE_NODE - Remove a node from the cluster

```
0000     1          .TITLE  CONMAN - Cluster Configuration Manager
0000     2          .IDENT  'V04-000'
0000     3 ;
0000     4 ;*****************************************************************************
0000     5 ;*                                                                           *
0000     6 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     7 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     8 ;*  ALL RIGHTS RESERVED.                                                     *
0000     9 ;*                                                                           *
0000    10 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    11 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    12 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    13 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    14 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    15 ;*  TRANSFERRED.                                                             *
0000    16 ;*                                                                           *
0000    17 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    18 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    19 ;*  CORPORATION.                                                             *
0000    20 ;*                                                                           *
0000    21 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    22 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    23 ;*                                                                           *
0000    24 ;*                                                                           *
0000    25 ;*****************************************************************************
0000    26 ;
0000    27
0000    28 ;++
0000    29 ; FACILITY: EXECUTIVE, CLUSTER MANAGEMENT
0000    30 ;
0000    31 ; ABSTRACT:
0000    32 ;       This module creates and manages the cluster configuration.  It is
0000    33 ;       responsible for forming a cluster, adding nodes to a cluster, and
0000    34 ;       reconfiguring a cluster following a failure.
0000    35 ;
0000    36 ; ENVIRONMENT: VAX/VMS
0000    37 ;
0000    38 ; AUTHOR: David W. Thiel,        CREATION DATE: 4-Apr-1983
0000    39 ;
0000    40 ; MODIFIED BY:
0000    41 ;
0000    42 ;       V03-022 DWT0225         David W. Thiel          11-Jul-1984
0000    43 ;               Change call to temporary name EXE$MNTVERSP2 to
0000    44 ;               real name EXE$CLUTRANIO.
0000    45 ;
0000    46 ;       V03-021 DWT0219         David W. Thiel          08-May-1984
0000    47 ;               Add synchronization between I/O and lock manager
0000    48 ;               clusters.
0000    49 ;
0000    50 ;       V03-020 DWT0213         David W. Thiel          17-Apr-1984
0000    51 ;               Always fork at least once when locking nodes to
0000    52 ;               begin a state transition.  Correct logic for
0000    53 ;               locking nodes when memory allocation fails.
0000    54 ;
0000    55 ;       V03-019 DWT0206         David W. Thiel          08-Apr-1984
0000    56 ;               Correct branch in CNX$RCVD_RECONFIG.
0000    57 ;               Tighten up timing on cluster formation to avoid
```

```
0000     58 ;               forming a paritioned cluster.  Add CNA$POWER_FAIL
0000     59 ;               entry point to receive control on power recovery.
0000     60 ;               Use CLUBFKB$V_FKB_BUSY instead of CLUB$V_FKB_BUSY.
0000     61 ;               Fix method of comparing quorum disk names from
0000     62 ;               various nodes.
0000     63 ;
0000     64 ;       V03-018 DWT0201         David W. Thiel          24-Mar-1984
0000     65 ;               Remove temporary code defining new symbols in the CLUB.
0000     66 ;
0000     67 ;       V03-017 DWT0194         David W. Thiel          23-Mar-1984
0000     68 ;               Change interface to disk quorum module to eliminate
0000     69 ;               window of ambiguity.  Rename CLUB$V_QF_SKIP_READ to
0000     70 ;               CLUB$V_QF_FAILED_NODE.  Update CSB$Q_REFTIME when a node
0000     71 ;               is added to or removed from the cluster.  Make sending
0000     72 ;               phase 1 messages a subroutine.  Make addition/deletion
0000     73 ;               of quorum disk from cluster a full two-phase transition.
0000     74 ;               Support requests for quorum change.  Remove propagation
0000     75 ;               of some bits to all nodes of cluster.
0000     76 ;
0000     77 ;       V03-016 DWT0187         David W. Thiel          5-Mar-1984
0000     78 ;               Correct and improve random number time-outs.
0000     79 ;
0000     80 ;       V03-015 DWT0181         David W. Thiel          27-Feb-1984
0000     81 ;               Correct MULL3 to be MULL2.
0000     82 ;
0000     83 ;       V03-014 DWT0173         David W. Thiel          27-Feb-1984
0000     84 ;               Start own reconfiguration attempts after some other
0000     85 ;               node tries and fails.  Raise interval to 0...20 seconds
0000     86 ;               when requesting cluster membership.  Correct use of
0000     87 ;               CLUB$W_QDVOTES to always minimize value and to propagate
0000     88 ;               when forming or adding nodes.  Add dummy entry
0000     89 ;               CNX$SHUTDOWN, update dummy CNX$CHANGE_QUORUM.
0000     90 ;
0000     91 ;       V03-013 DWT0161         David W. Thiel          03-Feb-1984
0000     92 ;               Modify use of CSB$B_NODEMAP to represent current
0000     93 ;               best estimate of global connection status.  Bit 0
0000     94 ;               is used for the quorum disk "connection".  This field
0000     95 ;               is maintained for use by the optimal sub-cluster
0000     96 ;               calculation.  Do optimal calculation on cluster
0000     97 ;               reconfigurations and cluster formation.  Always
0000     98 ;               copy CLUB$V_ACTIVE bit to CSB$V_ACTIVE bit in
0000     99 ;               CNX$DISK_CHANGE.  Correct blown register.
0000    100 ;
0000    101 ;       V03-012 DWT0147         David W. Thiel          19-Dec-1983
0000    102 ;               Add support for lock manager directory system vector.
0000    103 ;               Remove old lock manager directory system logic.
0000    104 ;               Add CLUB$W_MEMSEQ (cluster membership state sequence
0000    105 ;               number) and support for maintaining same.  Add use of
0000    106 ;               CLMPRO$L_FMERIT field and its propagation with phase
0000    107 ;               1 messages.  Use common routine CNX$QUORUM_CALC for
0000    108 ;               all quorum calculations.  Add support for a variable
0000    109 ;               number of votes for the quorum disk.  Remove filling
0000    110 ;               in several unused fields in the CLMSTS message.
0000    111 ;               Invoke fail-in table on all cluster membership
0000    112 ;               transitions.  Change name of CNX$BEGIN_FAILIN and
0000    113 ;               CNX$BEGIN_FAILOVER to CNX$MEMBERSHIP_CHANGE.
0000    114 ;               Use routines in new CONSUBS module.  Update TOPOLOGY
```

```
0000   115 ;                message logic.
0000   116 ;
0000   117 ;        V03-011 DWT0141         David W. Thiel          04-Nov-1983
0000   118 ;                Correct decision of who is to bugcheck when two
0000   119 ;                clusters meet.
0000   120 ;
0000   121 ;        V03-010 DWT0136         David W. Thiel          04-Oct-1983
0000   122 ;                Correct validation of transaction number in received
0000   123 ;                message to compare against the current transaction
0000   124 ;                rather than the maximum transaction number seen.
0000   125 ;                Use CLUB$V_LOST_CNX to mark situations where there
0000   126 ;                is a broken connection to a cluster member.
0000   127 ;
0000   128 ;        V03-009 DWT0132         David W. Thiel          23-Sep-1983
0000   129 ;                Correct misuse of register on an error path in
0000   130 ;                CNX$RCVD_DESC.  Improve state change messages to
0000   131 ;                be less repetitious.
0000   132 ;
0000   133 ;        V03-008 DWT0128         David W. Thiel          30-Aug-1983
0000   134 ;                Be less interested in received status when CLUB$V_INIT
0000   135 ;                flag has not yet been set -- avoid attempting to join
0000   136 ;                a cluster in this situation.
0000   137 ;                Enable writing quorum file when node is added to an
0000   138 ;                existing cluster.
0000   139 ;
0000   140 ;        V03-007 DWT0118         David W. Thiel          22-Aug-1983
0000   141 ;                Add quorum disk support.  Change CONFIG_CHANGE to
0000   142 ;                CNX$CONFIG_CHANGE.  Add logic to recover from a
0000   143 ;                failure while a cluster state transition is in
0000   144 ;                progress.  Add routines to process incoming QUORUM
0000   145 ;                and TOPOLOGY messages.
0000   146 ;
0000   147 ;        V03-006 DWT0116         David W. Thiel          1-Aug-1983
0000   148 ;                Change INIT_STD_RESP to CNX$INIT_STD_RESP and make
0000   149 ;                symbol gobal.  Change PROCESS_RESPONSE to
0000   150 ;                CNX$PROCESS_REPONSE and make symbol global.
0000   151 ;                Improve use of CLUB$L_TQE field.
0000   152 ;
0000   153 ;        V03-005 DWT0111         David W. Thiel          28-Jul-1983
0000   154 ;                Automatically raise quorum to be greater than half
0000   155 ;                of the sum of available votes.
0000   156 ;                Use real TQE for random timeouts.
0000   157 ;                Add CNX$CHANGE_QUORUM entry point.
0000   158 ;                Correct handling of status bits received from a
0000   159 ;                remote node.
0000   160 ;                Hang if no quorum using CNX$CHECK_QUORUM routine.
0000   161 ;
0000   162 ;        V03-004 DWT0109         David W. Thiel          25-Jun-1983
0000   163 ;                Select appropriate nodes when join message is received.
0000   164 ;                Defer setting directory system in joining node
0000   165 ;                until phase 2 message is received.
0000   166 ;
0000   167 ;        V03-003 DWT0108         David W. Thiel          23-Jun-1983
0000   168 ;                Fix broken BLBC instruction.  Fix logic for looking
0000   169 ;                up a node described in a message from a remote system.
0000   170 ;
0000   171 ;        V03-002 DWT0106         David W. Thiel          23-Jun-1983
```

```
0000    172 ;                   Correct search for CSB when node description received.
0000    173 ;                   Correct propagation of directory node when a node
0000    174 ;                   joins the cluster.
0000    175 ;                   Deallocate RSPID after use in DESCRIBE_NODE.
0000    176 ;                   When a connection breaks, disable locks and forbid
0000    177 ;                   additions of nodes to the cluster.
0000    178 ;                   Correct discrepancy in protocol for requesting membership
0000    179 ;                   in a cluster.
0000    180 ;
0000    181 ;           V03-001 DWT0103         David W. Thiel          27-May-1983
0000    182 ;                   Fix blown register.
0000    183 ;
0000    184 ;--
0000    185
```

```
                        .SBTTL  DECLARATIONS
                ;
                ; INCLUDE FILES:
                ;
                        $CDRPDEF                        ; CDRP offsets
                        $CDTDEF                         ; CDT Offsets
                        $CLMDRSDEF                      ; Cluster disconnect/reject codes
                        $CLSMSGDEF                      ; Cluster message definitions
                        $CLUBDEF                        ; CLUster Block offsets
                        $CSBDEF                         ; CSB Offsets
                        $DYNDEF                         ; Data structure type codes
                        $FKBDEF                         ; ForK Block offsets
                        $IPLDEF                         ; IPL definitions
                        $PCBDEF                         ; Process Control Block offsets
                        $PDTDEF                         ; Port Definition Table offsets
                        $SBDEF                          ; System Block offsets
                        $SSDEF                          ; Status code definitions
                        $TQEDEF                         ; TQE offsets

                ;
                ; OWN STORAGE:
                ;

00000000                .PSECT  $$$040,LONG             ; R/W Data PSECT

                ;*****************************************************************************
                ;
                ; NOTE: The following assumptions are in effect for this entire module.
                ;
                ;*****************************************************************************

                        ASSUME  IPL$_SYNCH  EQ  IPL$_SCS
                        ASSUME  IPL$_SYNCH  EQ  IPL$_TIMER

00000000                .PSECT  $$$100,LONG

                        .DEFAULT        DISPLACEMENT,WORD
```

```
                                0000   225          .SBTTL  CNX$CON_INIT - Initialization Call
                                0000   226
                                0000   227 ;++
                                0000   228 ;
                                0000   229 ; FUNCTIONAL DESCRIPTION:
                                0000   230 ;
                                0000   231 ;       This routine is called at system initialization to start up the
                                0000   232 ;       configuration manager.  This call must be made AFTER the CLUB
                                0000   233 ;       is created and initialized.  IPL is 31.
                                0000   234 ;
                                0000   235 ; CALLING SEQUENCE:
                                0000   236 ;
                                0000   237 ;       JSB     CNX$CON_INIT
                                0000   238 ;       IPL is 31
                                0000   239 ;
                                0000   240 ; INPUT PARAMETERS:
                                0000   241 ;
                                0000   242 ;       NONE
                                0000   243 ;
                                0000   244 ; OUTPUT PARAMETERS:
                                0000   245 ;
                                0000   246 ;       NONE
                                0000   247 ;
                                0000   248 ; COMPLETION CODES:
                                0000   249 ;
                                0000   250 ;       R0 contains status.
                                0000   251 ;
                                0000   252 ; SIDE EFFECTS:
                                0000   253 ;
                                0000   254 ;       R1 is destroyed.
                                0000   255 ;
                                0000   256 ;--
                                0000   257
                          00000000   258          .PSECT  $$$002,LONG             ; Initialization Psect
                                0000   259
                                0000   260 CNX$CON_INIT::
                           3C   BB     0000   261          PUSHR   #^M<R2,R3,R4,R5>        ; Save registers
                54  00000000'GF   D0     0002   262          MOVL    G^CLU$GL_CLUB,R4        ; Address of CLUB
                   00 1C A4    12   E2  0009   263          BBSS    #CLUB$V_NO_FORM, -      ; Set bit inhibiting cluster formation
                                0000   264                          CLUB$L_FLAGS(R4),10$
                      53   10 A4    D0  000E   265 10$:     MOVL    CLUB$L_LOCAL_CSB(R4),R3 ; Address of local CSB
            60 A3  02000008 8F    C8  0012   266          BISL2   #<CSB$M_STATUS_RCVD ! - ; Mark status received
                                0000   267                            CSB$M_QF_SAME>, -     ; Mark quorum files matching
                                0000   268                          CSB$L_STATUS(R3)        ;   for local CSB
                         51   18   3C  001A   269          MOVZWL  #FKB$K_LENGTH,R1        ; Fork block length
                            FFE0'   30  001D   270          BSBW    CNX$ALLOZMEM            ; Allocate and zero a block of memory
                           11   50   E9  0020   271          BLBC    R0,20$                  ; Branch on error
                         55   52     D0  0023   272          MOVL    R2,R5
                      0A A5   08     90  0026   273          MOVB    #DYN$C_FRK, -           ; Block type
                                0000   274                          FKB$B_TYPE(R5)
                      0B A5   08     90  002A   275          MOVB    #IPL$_SCS, -            ; Fork IPL
                                0000   276                          FKB$B_FIPL(R5)
                            FFCF'   30  002E   277          BSBW    INIT_CONTINUE           ; Put synchronous return on stack
                         50   01     D0  0031   278          MOVL    S^#SS$_NORMAL,R0        ; Set success status
                              3C     BA  0034   279 20$:     POPR    #^M<R2,R3,R4,R5>        ; Restore registers
                                     05  0036   280          RSB
                                0037   281
```

```
                        00000000    282          .PSECT  $$$100,LONG
                             0000   283
                             0000   284 INIT_CONTINUE:
            00000000'GF   16 0000   285          JSB     G^EXE$FORK              ; Fork
                             0006   286 ;
                             0006   287 ; Get here scon after IPL is lowered to SCS (8) at system initialization
                             0006   288 ;
                    FFF7'    30 0006   289          BSBW    CNX$RANDOM_INIT         ; Initialize random number generator
                             0009   290 ;
                             0009   291 ; Wait for SYSINIT to start up and declare us ready for think about forming a cluste
                             0009   292 ;
                    0269     30 0009   293 10$:     BSBW    CNX$SEND_ALL_STATUS     ; Send status as necessary
                             000C   294          FORK_WAIT                       ; Wait a second
                F2 1C A4      13 E1 0012   295          BBC     #CLUB$V_INIT, -         ; Branch if SYSINIT has not yet triggered us
                             0017   296                  CLUB$L_FLAGS(R4),10$
                       30    BB 0017   297          PUSHR   #^M<R4,R5>              ; Save registers
                    0539     30 0019   298          BSBW    SCAN                    ; Look for work to do
                       30    BA 001C   299          POPR    #^M<R4,R5>              ; Restore registers
                             001E   300 ;
                             001E   301 ; We must now delay an appropriate time before forming a cluster in order to allow
                             001E   302 ; the discovery of other nodes that are already cluster members before attempting
                             001E   303 ; to form our own cluster.  This time is the larger of:
                             001E   304 ;       a) SCS process poller interval * largest PDT polling sweep interval
                             001E   305 ;       b) Time to find quorum disk device and acquire a "connection" to the
                             001E   306 ;               quorum disk
                             001E   307 ; In fact, we ignore the latter at this time.
                             001E   308 ; We do the wait in two steps.  First we wait the process poller interval because
                             001E   309 ; this is known.  Then we wait a time which is the maximum of the polling sweep
                             001E   310 ; intervals of all ports that have been found during the earlier interval.
                             001E   311 ;
          53    00000000'GF   3C 001E   312          MOVZWL  G^SCS$GW_PRCPOLINT,R3   ; Process polling interval
                             0025   313 20$:     FORK_WAIT                       ; Wait a second
             32 1C A4      00 E0 002B   314          BBS     #CLUB$V_CLUSTER, -      ; Branch if a cluster has been formed
                             0030   315                  CLUB$L_FLAGS(R4),60$
                      F2 53    F4 0030   316          SOBGEQ  R3,20$                  ; Wait for interval to complete
                             0033   317          ;
                             0033   318          ; By now, all SCS ports should have been found.
                             0033   319          ; Scan list of all PDT's computing the largest value of PDT$L_POLLSWEEP that
                             0033   320          ; can be found.  This is the interval that we should now wait before
                             0033   321          ; forming a cluster, i.e. it is the interval that may elapse before a
                             0033   322          ; connection may be made to an existing cluster node.
                             0033   323          ;
                          53 D4 0033   324          CLRL    R3                      ; Accumulator for larger PDT$L_POLLSWEEP
          50    00000000'GF   DE 0035   325          MOVAL   G^SCS$GL_PDT,R0         ; Get address of PDT listhead
                       50   60 D0 003C   326 30$:     MOVL    (R0),R0                 ; Link to next PDT
                            0E 13 003F   327          BEQL    50$                     ; Branch when all done
                    53   00D8 C0 D1 0041   328          CMPL    PDT$L_POLLSWEEP(R0),R3  ; Is this value larger?
                            05 1B 0046   329          BLEQU   40$                     ; Branch if no
                    53   00D8 C0 D0 0048   330          MOVL    PDT$L_POLLSWEEP(R0),R3  ; Update largest seen yet
                            ED 11 004D   331 40$:     BRB     30$                     ; Continue iteration
                             004F   332
                             004F   333 50$:     FORK_WAIT                       ; Wait a second
             08 1C A4      00 E0 0055   334          BBS     #CLUB$V_CLUSTER, -      ; Branch if a cluster has been formed
                             005A   335                  CLUB$L_FLAGS(R4),60$
                      F2 53    F4 005A   336          SOBGEQ  R3,50$                  ; Wait for interval to complete
             00 1C A4      12 E5 005D   337          BBCC    #CLUB$V_NO_FORM, -      ; Clear formation inhibit
                             0062   338                  CLUB$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   343          .SBTTL  CNX$POWER_FAIL - Power Fail Recovery Entry
                                006E   344
                                006E   345 ;++
                                006E   346 ;
                                006E   347 ; FUNCTIONAL DESCRIPTION:
                                006E   348 ;
                                006E   349 ;       This routine is called during recovery from a power failre.
                                006E   350 ;       IPL is 31.
                                006E   351 ;
                                006E   352 ; CALLING SEQUENCE:
                                006E   353 ;
                                006E   354 ;       JSB     CNX$POWER_FAIL
                                006E   355 ;       IPL is 31
                                006E   356 ;
                                006E   357 ; INPUT PARAMETERS:
                                006E   358 ;
                                006E   359 ;       NONE
                                006E   360 ;
                                006E   361 ; OUTPUT PARAMETERS:
                                006E   362 ;
                                006E   363 ;       NONE
                                006E   364 ;
                                006E   365 ; COMPLETION CODES:
                                006E   366 ;
                                006E   367 ;       NONE
                                006E   368 ;
                                006E   369 ; SIDE EFFECTS:
                                006E   370 ;
                                006E   371 ;       All registers are preserved
                                006E   372 ;
                                006E   373 ;--
                                006E   374
                                006E   375 CNX$POWER_FAIL::
                          38    BB    006E   376          PUSHR   #^M<R3,R4,R5>           ; Save registers
54   00000000'GF              D0    0070   377          MOVL    G^CLU$GL_CLUB,R4       ; Address of CLUB
     55    018C C4             9E    0077   378          MOVAB   CLUB$B_CLUBPWF(R4),R5  ; Fork block address
  02 18 A5    00              E2    007C   379          BBSS    #CLUBPWF$V_BUSY, -     ; Branch if block is in use
                                0081   380                  CLUBPWF$L_STATUS(R5),10$ ;   by a previous power recovery
                          03    10    0081   381          BSBB    20$                    ; Save synchronous return address
                          38    BA    0083   382 10$:     POPR    #^M<R3,R4,R5>          ; Restore registers
                                05    0085   383          RSB                            ; Return to power fail notification code
                                0086   384
   00000000'GF                16    0086   385 20$:     JSB     G^EXE$FORK             ; Fork to IPL SYNCH, SCS
                                008C   386 ;
                                008C   387 ; Continue here at IPL 8
                                008C   388 ;
  20 18 A5    00              E5    008C   389          BBCC    #CLUBPWF$V_BUSY, -     ; Clear busy bit, branch on error
                                0091   390                  CLUBPWF$L_STATUS(R5),60$
  00 1C A4    1E              E5    0091   391          BBCC    #CLUB$V_QF_DYNVOTE, -  ; Clear bit counting disk in dynamic quorum
                                0096   392                  CLUB$L_FLAGS(R4),30$
  00 1C A4    18              E2    0096   393 30$:     BBSS    #CLUB$V_QF_FAILED_NODE, - ; Trigger quorum disk manager to reacquire
                                009B   394                  CLUB$L_FLAGS(R4),40$
     52   1C A4                D0    009B   395 40$:     MOVL    CLUB$L_FLAGS(R4),R2    ; Save flags
          FF5E'                 30    009F   396          BSBW    CNX$CHECK_QUORUM       ; Check for out of quorum condition
     52   1C A4                CA    00A2   397          BICL2   CLUB$L_FLAGS(R4),R2    ; Set flag in R2 if clear now and set earlie
     06 52    1C                E0    00A6   398          BBS     #CLUB$V_QUORUM,R2,50$  ; Branch if quorum was just lost and mntver
                                00AA   399                                             ;   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   408          .SBTTL  CNX$CON_NEWSYS - Connected to New System
                        00B5   409
                        00B5   410 ;++
                        00B5   411 ;
                        00B5   412 ; FUNCTIONAL DESCRIPTION:
                        00B5   413 ;
                        00B5   414 ;       This routine is called when a connection to a new system is
                        00B5   415 ;       made.  It will exchange status with the new system and consider
                        00B5   416 ;       forming a cluster, joining the cluster of which the new system is
                        00B5   417 ;       a member, or wait for the new system to join the cluster of which
                        00B5   418 ;       this system is a member.
                        00B5   419 ;
                        00B5   420 ; CALLING SEQUENCE:
                        00B5   421 ;
                        00B5   422 ;       JSB     CNX$CON_NEWSYS
                        00B5   423 ;       IPL is IPL$_SCS
                        00B5   424 ;
                        00B5   425 ; INPUT PARAMETERS:
                        00B5   426 ;
                        00B5   427 ;       R5: CSB of new system
                        00B5   428 ;
                        00B5   429 ; OUTPUT PARAMETERS:
                        00B5   430 ;
                        00B5   431 ;       NONE
                        00B5   432 ;
                        00B5   433 ; COMPLETION CODES:
                        00B5   434 ;
                        00B5   435 ;       NONE
                        00B5   436 ;
                        00B5   437 ; SIDE EFFECTS:
                        00B5   438 ;
                        00B5   439 ;       R0 and R1 are destroyed
                        00B5   440 ;
                        00B5   441 ;--
                        00B5   442
                        00B5   443 CNX$CON_NEWSYS::
                  3C    BB    00B5   444          PUSHR   #^M<R2,R3,R4,R5>        ; Save registers
                  53    55    D0     00B7   445          MOVL    R5,R3           ; Put CSB address in R3
00 60 A3          1A    E3    00BA   446          BBCS    #CSB$V_SEND_STATUS, -   ; Need to send status
                              00BF   447                  CSB$L_STATUS(R3),10$
                  03F4  30    00BF   448 10$:     BSBW    START_TIMEOUT           ; Start CLUB-based timer
                  3C    BA    00C2   449          POPR    #^M<R2,R3,R4,R5>        ; Restore registers
                        05    00C4   450          RSB                             ; Return
```

```
                                00C5    452          .SBTTL  CNX$CON_BREAK - Connection has broken
                                00C5    453
                                00C5    454 ;++
                                00C5    455 ;
                                00C5    456 ; FUNCTIONAL DESCRIPTION:
                                00C5    457 ;
                                00C5    458 ;       This routine is called when a connection to a system has broken and
                                00C5    459 ;       attempts to reclose the connection have failed.
                                00C5    460 ;
                                00C5    461 ; CALLING SEQUENCE:
                                00C5    462 ;
                                00C5    463 ;       JSB     CNX$CON_BREAK
                                00C5    464 ;       IPL is IPL$_SCS
                                00C5    465 ;
                                00C5    466 ; INPUT PARAMETERS:
                                00C5    467 ;
                                00C5    468 ;       R5: CSB of system to which the connection is lost
                                00C5    469 ;
                                00C5    470 ; OUTPUT PARAMETERS:
                                00C5    471 ;
                                00C5    472 ;       NONE
                                00C5    473 ;
                                00C5    474 ; COMPLETION CODES:
                                00C5    475 ;
                                00C5    476 ;       NONE
                                00C5    477 ;
                                00C5    478 ; SIDE EFFECTS:
                                00C5    479 ;
                                00C5    480 ;       R0 and R1 are destroyed
                                00C5    481 ;
                                00C5    482 ;--
                                00C5    483
                                00C5    484 CNX$CON_BREAK::
                          3C    BB    00C5    485          PUSHR   #^M<R2,R3,R4,R5>        ; Save registers
                      53  55    D0    00C7    486          MOVL    R5,R3                   ; Put CSB address in R3
                  54    64 A3   D0    00CA    487          MOVL    CSB$L_CLUB(R3),R4       ; Address of CLUB
              55    010C C4     9E    00CE    488          MOVAB   CLUB$B_CLUFCB(R4),R5    ; Address of CLUFCB
                                00D3    489
           22 60 A3     01      E1    00D3    490          BBC     #CSB$V_MEMBER, -        ; Branch if lost node is
                                00D8    491                  CSB$L_STATUS(R3),10$    ;   not a cluster member
    1C A4    00800000 8F      C8    00D8    492          BISL2   #CLUB$M_LOST_CNX, -     ; Mark connection to member lost
                                00E0    493                  CLUB$L_FLAGS(R4)
       00000000'GF   01      8E    00E0    494          MNEGB   #1,G^LCK$GB_STALLREQS   ; Stall lock requests
             50    4C A3     3C    00E7    495          MOVZWL  CSB$W_CSID_IDX(R3),R0   ; CSID index value
          00 28 A5    50      E5    00EB    496          BBCC    R0, -                   ; Clear participation bit in
                                00F0    497                  CLUFCB$B_NODEMAP(R5),5$ ;  failover table for lost node
             51    10 A4     D0    00F0    498 5$:      MOVL    CLUB$L_LOCAL_CSB(R4),R1 ; Local node CSB address
        00 008C C1    50      E5    00F4    499          BBCC    R0,CSB$B_NODEMAP(R1),10$ ; Mark loss of connectivity from local node
                                00FA    500 10$:
           32 1C A4    1D      E1    00FA    501          BBC     #CLUB$V_TRANSITION, -   ; Branch if no state change in
                                00FF    502                  CLUB$L_FLAGS(R4), 60$   ;   progress
           10 A4    5C A4      D1    00FF    503          CMPL    CLUB$L_COORD(R4), -     ; Is this node the coordinator
                                0104    504                  CLUB$L_LOCAL_CSB(R4)
                       1F      13    0104    505          BEQL    40$                     ; Branch if this node is the coordinator
                                0106    506 ;
                                0106    507 ; Here if transition is in progress and this node is not the coordinator
                                0106    508 ;
```

```
                53   5C A4   D1   0106    509          CMPL    CLUB$L_COORD(R4),R3     ; Is connection to coordinator lost?
                        28   12   010A    510          BNEQ    70$                     ; Branch if not coordinator connection
                                  010C    511 ;
                                  010C    512 ; The connection to the coordinator has been lost and this node is in the midst of
                                  010C    513 ; a cluster state transition.  If a Phase 1 message has not yet been received, it is
                                  010C    514 ; safe to back out the transition because we know that noone can have seen a Phase 2
                                  010C    515 ; message.
                                  010C    516 ;
        40 8F    59 A4   91   010C    517          CMPB    CLUB$B_CUR_PHASE(R4), - ; Has Phase 1 been seen yet?
                                  0111    518                  #CLMCNX$K_PH1
                        0D   1F   0111    519          BLSSU   30$                     ; Branch if no Phase 1 yet
                                  0113    520 ;
                                  0113    521 ; This node has received a Phase 1 message.  If this node is not a member of a
                                  0113    522 ; cluster, then bugcheck -- we can't cleanly resolve what might be happening.
                                  0113    523 ;
     04 1C A4    00   E1   0113    524          BBC     #CLUB$V_CLUSTER, -      ; Branch if not a cluster member and
                                  0118    525                  CLUB$L_FLAGS(R4),20$    ;   resolve the situation the easy way
                        1D   10   0118    526          BSBB    ANALYZE_PHASE           ; Make complex decision about what to do
                        18   11   011A    527          BRB     70$
                                  011C    528 ;
                                  011C    529 ; This node is not a member of a cluster, is in Phase 1 of a cluster state transitio
                                  011C    530 ; and has lost the connection to the transition coordinator.  It is not trivial
                                  011C    531 ; to resolve this situation.  Take the easy way out.
                                  011C    532 ;
                                  011C    533 20$:    BUG_CHECK       CLUEXIT,FATAL   ; Can't handle this failure case
                                  0120    534 ;
                      1123   30   0120    535 30$:    BSBW    UNLOCK_ALL              ; Unlock things
                        0F   11   0123    536          BRB     70$
                                  0125    537
                                  0125    538 40$:
                                  0125    539 ;
                                  0125    540 ; Here if transition is in progress and this node is the coordinator
                                  0125    541 ;
     60 8F    59 A4   91   0125    542          CMPB    CLUB$B_CUR_PHASE(R4), - ; Has Phase 2 been sent yet?
                                  012A    543                  #CLMCNX$K_PH2
                        05   1E   012A    544          BGEQU   50$                     ; Branch if phase 2 sent
     00 1C A4    11   E3   012C    545          BBCS    #CLUB$V_UNLOCK, -       ; No phase 2 sent -- request
                                  0131    546                  CLUB$L_FLAGS(R4), 50$   ;   unlock
                                  0131    547 50$:
                                  0131    548 ;
                                  0131    549 ; Common exit routine
                                  0131    550 ;
                      0421   30   0131    551 60$:    BSBW    SCAN                    ; Start CLUB-based timer
                        3C   BA   0134    552 70$:    POPR    #^M<R2,R3,R4,R5>        ; Restore registers
                             05   0136    553          RSB                             ; Return
```

```
0137    555          .SBTTL  ANALYZE_PHASE - Determine Transition Phase and Finish Transition
0137    556
0137    557 ;++
0137    558 ;
0137    559 ; FUNCTIONAL DESCRIPTION:
0137    560 ;
0137    561 ;       This routine is called when all of the following are true:
0137    562 ;       1)      The local node is a member of a cluster
0137    563 ;       2)      A cluster state transition is in progress
0137    564 ;       3)      The connection to the coordinator node has broken
0137    565 ;       4)      A Phase 1 message has been received
0137    566 ;
0137    567 ;       This routine must determine whether to commit or backout the transition.
0137    568 ;
0137    569 ; CALLING SEQUENCE:
0137    570 ;
0137    571 ;       JSB     ANALYZE_PHASE
0137    572 ;       IPL is IPL$_SCS
0137    573 ;       The caller is returned to synchronously, but the function performed
0137    574 ;       continues asynchronously and no notification is given to the caller
0137    575 ;       upon completion.
0137    576 ;
0137    577 ; INPUT PARAMETERS:
0137    578 ;
0137    579 ;       R3: CSB of system to which the connection is lost
0137    580 ;       R4: Address of CLUB
0137    581 ;
0137    582 ; OUTPUT PARAMETERS:
0137    583 ;
0137    584 ;       NONE
0137    585 ;
0137    586 ; COMPLETION CODES:
0137    587 ;
0137    588 ;       NONE
0137    589 ;
0137    590 ; SIDE EFFECTS:
0137    591 ;
0137    592 ;       R0 through R5 are destroyed
0137    593 ;
0137    594 ; ALGORITHM:
0137    595 ;
0137    596 ;       The strategy here is to poll all remaining cluster members and
0137    597 ;       see whether they can resolve out dilemma.
0137    598 ;       If another member has:
0137    599 ;               Not yet seen Phase 1 ==> Undo the transition in progress
0137    600 ;               Already seen Phase 2 ==> Commit the transition in progress
0137    601 ;               Seen Phase 1 and has a coordinator connection ==> Restart poll
0137    602 ;               Seen Phase 1 and has no coordinator connection ==> Continue poll
0137    603 ;       If this completes without yield a definitive (Undo or Commit) result,
0137    604 ;       poll all nodes again.  If another member has:
0137    605 ;               Not yet seen Phase 1 ==> Undo the transition in progress
0137    606 ;               Already seen Phase 2 ==> Commit the transition in progress
0137    607 ;               Phase 1 and has a coordinator connection ==> inconsistent with above
0137    608 ;               Phase 1 and has no coordinator connection ==>
0137    609 ;                       Obtain promise from node that it will never commit this
0137    610 ;                       transition
0137    611 ;                       Continue poll
```

```
                                0137   612 ;       If this completes without yielding a definitive (Undo or Commit) result,
                                0137   613 ;       do not commit the transition.
                                0137   614 ;
                                0137   615 ;--
                                0137   616
                                0137   617 ANALYZE_PHASE:
1C A4    00100000 8F    CA      0137   618           BICL2   #CLUB$M_BACKOUT, -      ; Clear backout inhibit bit
                                013F   619           CLUB$L_FLAGS(R4)
1C A4    00000100 8F    CA      013F   620 10$:      BICL2   #CLUB$M_STS_PPHASE, -  ; Clear polling phase flag (Phase A)
                                0147   621           CLUB$L_FLAGS(R4)
1C A4    00001E00 8F    CA      0147   622 20$:      BICL2   #< -                   ; Clear poll result bits
                                014F   623           CLUB$M_STS_PH0 ! -             ;   phase 0 seen
                                014F   624           CLUB$M_STS_PH1 ! -             ;   phase 1 seen (good coordinator connectio
                                014F   625           CLUB$M_STS_PH1B ! -            ;   phase 1 seen (broken coordinator connect
                                014F   626           CLUB$M_STS_PH2>, -             ;   phase 2 seen
                                014F   627           CLUB$L_FLAGS(R4)
              5A A4    B4      014F   628           CLRW    CLUB$W_MSGCNT(R4)      ; Initialize waiting count
               FEAB'   30      0152   629           BSBW    CNX$SCAN_CSBS          ; Iterate over all CSBs
              4D 50    E9      0155   630           BLBC    R0,70$                 ; Branch when done
        18 60 A3  01    E1      0158   631           BBC     #CSB$V_MEMBER, -       ; Branch if not a member
                                015D   632           CSB$L_STATUS(R3),30$
        13 60 A3  00    E0      015D   633           BBS     #CSB$V_LONG_BREAK, -   ; Branch if permanent break
                                0162   634           CSB$L_STATUS(R3),30$   ;   in connection
        0E 60 A3  18    E0      0162   635           BBS     #CSB$V_LOCAL, -        ; Branch if local CSB
                                0167   636           CSB$L_STATUS(R3),30$
               FE96'   30      0167   637           BSBW    CNX$ALLOC_WARMCDRP_CSB
              09 50    E9      016A   638           BLBC    R0,40$                 ; No memory available
              5A A4    B6      016D   639           INCW    CLUB$W_MSGCNT(R4)      ; Include in wait count
                 07    10      0170   640           BSBB    50$                    ; Interogate node
               FE8B'   31      0172   641           BRW     CNX$SCAN_CSBS_FORK     ; Release control mementarily
                               0175   642
                 05    0175   643 30$:      RSB                                    ; Return and continue scan
                               0176   644
               FE87'   31      0176   645 40$:      BRW     CNX$SCAN_CSBS_RETRY    ; Delay and resume scan
                               0179   646 ;
                               0179   647 ; Send status inquiry to one node
                               0179   648 ;
           50    0E    9A      0179   649 50$:      MOVZBL  #CLMCNX$K_FNC_TRNSTS,R0 ; Facility specific message code
     04 1C A4    08    E1      017C   650           BBC     #CLUB$V_STS_PPHASE, -  ; Branch if status inquiry phase A
                               0181   651           CLUB$L_FLAGS(R4),60$
           50    18    E2      0181   652           BBSS    #CLMCNX$V_RP_TRNSTS_CMT+24, - ; Request commitment on PH1B status
                 00            0184   653           R0,60$                 ;   -- to go into message reply field
              118A   30      0185   654 60$:      BSBW    INIT_STD_MSG           ; Standard CDRP message initialization
               FE75'   30      0188   655           BSBW    CNX$SEND_MSG_CSB       ; Send message
                               018B   656 ;
                               018B   657 ;       We are resumed here when the response message arrives.
                               018B   658 ;       Registers contain:
                               018B   659 ;               R0:     Status
                               018B   660 ;               R2:     Address of message buffer
                               018B   661 ;               R3:     Address of CSB
                               018B   662 ;               R4:     Address of PDT
                               018B   663 ;               R5:     Address of CDRP
                               018B   664 ;
              1158   30      018B   665           BSBW    CNX$PROCESS_RESPONSE   ; Deallocate storage, get status
             14 50    E9      018E   666           BLBC    R0,70$                 ; Branch if NACK or connection broken
                               0191   667           ASSUME  CLMCNX$K_RP_TRNSTS_PH0+1 EQ CLMCNX$K_RP_TRNSTS_PH1B
                               0191   668           ASSUME  CLMCNX$K_RP_TRNSTS_PH1B+1 EQ CLMCNX$K_RP_TRNSTS_PH1
```

```
                     0191   669          ASSUME  CLMCNX$K_RP_TRNSTS_PH1+1 EQ CLMCNX$K_RP_TRNSTS_PH2
                     0191   670          ASSUME  CLUB$V_STS_PH0+1 EQ CLUB$V_STS_PH1B
                     0191   671          ASSUME  CLUB$V_STS_PH1B+1 EQ CLUB$V_STS_PH1
                     0191   672          ASSUME  CLUB$V_STS_PH1+1 EQ CLUB$V_STS_PH2
   50   13 A2  9A    0191   673          MOVZBL  CLMCNX$B_REPLY(R2),R0
      50   01  C2    0195   674          SUBL2   #CLMCNX$K_RP_TRNSTS_PH0,R0 ; Base on zero
      04   50  D1    0198   675          CMPL    R0,#CLMCNX$K_RP_TRNSTS_PH2 ; Check range
           0E  1A    019B   676          BGTRU   80$                        ; Out of range reply, die
      50   09  C0    019D   677          ADDL2   #CLUB$V_STS_PH0,R0         ; Compute bit number in CLUB
00 1C A4   50  E2    01A0   678          BBSS    R0,CLUB$L_FLAGS(R4),70$ ; Set status bit for this node
                     01A5   679 ;
                     01A5   680 ; terminate threads -- except for the last one
                     01A5   681 ;
      5A A4  B7    01A5   682 70$:     DECW    CLUB$W_MSGCNT(R4)
         05  19    01A8   683          BLSS    90$                        ; Branch when done
             05    01AA   684          RSB                                ; Terminate thread
                     01AB   685
                     01AB   686 80$:     BUG_CHECK       CNXMGRERR,FATAL ; Inconsistent message data received
                     01AF   687 ;
                     01AF   688 ; Get here after all available member nodes have been polled.
                     01AF   689 ; Evaluate results of the poll and take appropriate action
                     01AF   690 ;
   50   1C A4  D0    01AF   691 90$:     MOVL    CLUB$L_FLAGS(R4),R0        ; Get flag bits for easy handling
08 50     09  E1    01B3   692          BBC     #CLUB$V_STS_PH0,R0,100$ ; Branch if Phase 0 not seen
04 50     0C  E1    01B7   693          BBC     #CLUB$V_STS_PH2,R0,100$ ; Branch if Phase 2 not seen
        FE42  30    01BB   694          BSBW    CNX$BUGCHECK_CLUSTER       ; Phase 0 and Phase 2 were seen -- die!
             05    01BE   695          RSB
                     01BF   696
   03 50     09  E1    01BF   697 100$:    BBC     #CLUB$V_STS_PH0,R0,110$ ; Branch if Phase 0 not seen
           1080  31    01C3   698          BRW     UNLOCK_ALL                 ; Some node was in Phase 0 -- Bail out and r
                     01C6   699
   0B 50     0C  E1    01C6   700 110$:    BBC     #CLUB$V_STS_PH2,R0,130$ ; Branch if Phase 2 not seen
   03 50     14  E0    01CA   701          BBS     #CLUB$V_BACKOUT,R0,120$ ; Branch if we have commited to backing out
           0F37  31    01CE   702          BRW     DO_PHASE2                  ; Pretend a phase 2 message has arrived
                     01D1   703
                     01D1   704 120$:    BUG_CHECK       CLUEXIT,FATAL   ; Saw phase 2 after commiting to backing out
                     01D5   705
0A 1C A4     08  E2    01D5   706 130$:    BBSS    #CLUB$V_STS_PPHASE, -      ; Set polling phase B and do it
                     01DA   707                  CLUB$L_FLAGS(R4),150$
                     01DA   708 ;
                     01DA   709 ; Polling phase A just completed
                     01DA   710 ;
   03 50     0B  E1    01DA   711          BBC     #CLUB$V_STS_PH1,R0,140$ ; Phase 1 with good connection not seen
           FF5E  31    01DE   712          BRW     10$                        ; Repeat first polling phase
           FF63  31    01E1   713 140$:    BRW     20$                        ; Perform second polling phase
                     01E4   714
   04 50     0B  E1    01E4   715 150$:    BBC     #CLUB$V_STS_PH1,R0,160$ ; Phase 1 with good connection not seen
                     01E8   716          BUG_CHECK       CNXMGRERR,FATAL ; Consistency error -- node with good connec
                     01EC   717                                          ; seen during polling phase B
                     01EC   718
           1057  31    01EC   719 160$:    BRW     UNLOCK_ALL                 ; Second phase was unconclusive, back-out
```

```
                        01EF    721          .SBTTL  CNX$DISK_CHANGE - Quorum Disk Connection State Change
                        01EF    722
                        01EF    723 ;++
                        01EF    724 ;
                        01EF    725 ; FUNCTIONAL DESCRIPTION:
                        01EF    726 ;
                        01EF    727 ;       This routine is called when the "connection" to the quorum disk has broken
                        01EF    728 ;       or has been re-established.
                        01EF    729 ;
                        01EF    730 ; CALLING SEQUENCE:
                        01EF    731 ;
                        01EF    732 ;       JSB     CNX$DISK_CHANGE
                        01EF    733 ;       Kernel mode, IPL .le. IPL$_SCS = IPL$_SYNCH
                        01EF    734 ;
                        01EF    735 ; INPUT PARAMETERS:
                        01EF    736 ;
                        01EF    737 ;       NONE
                        01EF    738 ;
                        01EF    739 ; OUTPUT PARAMETERS:
                        01EF    740 ;
                        01EF    741 ;       NONE
                        01EF    742 ;
                        01EF    743 ; COMPLETION CODES:
                        01EF    744 ;
                        01EF    745 ;       NONE
                        01EF    746 ;
                        01EF    747 ; SIDE EFFECTS:
                        01EF    748 ;
                        01EF    749 ;       R0 and R1 are destroyed
                        01EF    750 ;
                        01EF    751 ;--
                        01EF    752
                        01EF    753 CNX$DISK_CHANGE::
                        01EF    754          DSBINT  #IPL$_SYNCH                ; Save and raise IPL
                     3C BB      01F5    755          PUSHR   #^M<R2,R3,R4,R5>           ; Save registers
                54   00000000'GF D0      01F7    756          MOVL    G^CLU$GL_CLUB,R4   ; Address of CLUB
                   53   10 A4   D0      01FE    757          MOVL    CLUB$L_LOCAL_CSB(R4),R3 ; Address of local CSB
           50   1C A4   01   01 EF      0202    758          EXTZV   #CLUB$V_QF_ACTIVE,#1, - ; Extract quorum file active bit
                                        0208    759                  CLUB$L_FLAGS(R4),R0     ;   from CLUB
           60 A3   01   09   50 F0      0208    760          INSV    R0, -              ; Insert quorum file active bit
                                        020E    761                  #CSB$V_QF_ACTIVE,#1, -  ;   extracted from CLUB into the
                                        020E    762                  CSB$L_STATUS(R3)        ;   local CSB
                        FDEF'   30      020E    763          BSBW    CNX$SCAN_CSBS              ; Iterate over all CSBs
                     06 50      E9      0211    764          BLBC    R0,30$                     ; Branch when done
                00 60 A3   1A   E2      0214    765          BBSS    #CSB$V_SEND_STATUS, -      ; Set status request bit
                                        0219    766                  CSB$L_STATUS(R3),20$
                        05      0219    767 20$:     RSB
                                        021A    768
                        FDE3'   30      021A    769 30$:     BSBW    CNX$CHECK_QUORUM           ; Hang if necessary
                        0335    30      021D    770          BSBW    SCAN                       ; Look for work to do
                          3C    BA      0220    771          POPR    #^M<R2,P3,R4,R5>           ; Restore registers
                                        0222    772          ENBINT                             ; Restore IPL
                        05      0225    773          RSB                                ; Return
```

```
                                0226   775          .SBTTL  CNX$CHANGE_QUORUM - Adjust quorum
                                0226   776
                                0226   777 ;++
                                0226   778 ;
                                0226   779 ; FUNCTIONAL DESCRIPTION:
                                0226   780 ;
                                0226   781 ;       Request that the cluster quorum be updated to a
                                0226   782 ;       specified value.  This requires a full cluster state
                                0226   783 ;       transition to agree on a new value.  The value is
                                0226   784 ;       adjusted so as not to leave the cluster without a
                                0226   785 ;       quorum and so as not to risk paritioning given a
                                0226   786 ;       universe of the nodes currently in the cluster.
                                0226   787 ;
                                0226   788 ; CALLING SEQUENCE:
                                0226   789 ;
                                0226   790 ;       JSB     CNX$CHANGE_QUORUM
                                0226   791 ;       IPL is IPL$_SCS
                                0226   792 ;
                                0226   793 ;       The requestor can determine that the operator is complete by
                                0226   794 ;       polling the CLUB$V_ADJ_QUORUM bit in the CLUB$L_FLAGS word.
                                0226   795 ;       This bit is cleared after the quorum has been adjusted.
                                0226   796 ;
                                0226   797 ; INPUT PARAMETERS:
                                0226   798 ;
                                0226   799 ;       R1 is new quorum value
                                0226   800 ;
                                0226   801 ; OUTPUT PARAMETERS:
                                0226   802 ;
                                0226   803 ;       NONE
                                0226   804 ;
                                0226   805 ; COMPLETION CODES:
                                0226   806 ;
                                0226   807 ;       R0 is success status.
                                0226   808 ;
                                0226   809 ; SIDE EFFECTS:
                                0226   810 ;
                                0226   811 ;       R1 is destroyed.
                                0226   812 ;
                                0226   813 ;--
                                0226   814
                                0226   815 CNX$CHANGE_QUORUM::
                      3C   BB   0226   816          PUSHR   #^M<R2,R3,R4,R5>        ; Save registers
                      50   D4   0228   817          CLRL    R0                      ; Assume failure
54   00000000'GF       D0   022A   818          MOVL    G^CLU$GL_CLUB,R4        ; Address of CLUB
   10 1C A4    1D   E0   0231   819          BBS     #CLUB$V_TRANSITION, -   ; Branch if a state change in progress
                         0236   820                  CLUB$L_FLAGS(R4),20$
   0B 1C A4    1B   E2   0236   821          BBSS    #CLUB$V_ADJ_QUORUM, -   ; Set quorum update request bit
                         023B   822                  CLUB$L_FLAGS(R4),20$    ;   branch if already set
    00A6 C4    51   B0   023B   823          MOVW    R1, -                   ; Store requested quorum value
                         0240   824                  CLUB$W_ADJ_QUORUM(R4)
            0312   30   0240   825 10$:      BSBW    SCAN                    ; Look for work to do
          50    01   3C   0243   826          MOVZWL  #SS$_NORMAL,R0          ; Return success status
                3C   BA   0246   827 20$:      POPR    #^M<R2,R3,R4,R5>        ; Restore registers
                     05   0248   828          RSB                             ; Return
```

```
                                0249   830          .SBTTL  CNX$SHUTDOWN - Request cluster shutdown
                                0249   831
                                0249   832 ;++
                                0249   833 ;
                                0249   834 ; FUNCTIONAL DESCRIPTION:
                                0249   835 ;
                                0249   836 ;       This entry point is called to request a cluster shutdown.
                                0249   837 ;       While in shutdown mode:
                                0249   838 ;               No nodes may join cluster.
                                0249   839 ;               If quorum is lost, shutdown immediately.
                                0249   840 ;               When all systems reach shutdown state, shutdown this node.
                                0249   841 ;
                                0249   842 ; CALLING SEQUENCE:
                                0249   843 ;
                                0249   844 ;       JSB     CNX$SHUTDOWN
                                0249   845 ;       IPL is IPL$_SCS
                                0249   846 ;
                                0249   847 ; INPUT PARAMETERS:
                                0249   848 ;
                                0249   849 ;       NONE
                                0249   850 ;
                                0249   851 ; OUTPUT PARAMETERS:
                                0249   852 ;
                                0249   853 ;       NONE
                                0249   854 ;
                                0249   855 ; COMPLETION CODES:
                                0249   856 ;
                                0249   857 ;       R0 is success status
                                0249   858 ;
                                0249   859 ; SIDE EFFECTS:
                                0249   860 ;
                                0249   861 ;       R1 is destroyed.
                                0249   862 ;
                                0249   863 ;--
                                0249   864
                                0249   865 CNX$SHUTDOWN::
                          3C    BB    0249   866          PUSHR   #^M<R2,R3,R4,R5>          ; Save registers
54    00000000'GF             D0    024B   867          MOVL    G^CLU$GL_CLUB,R4          ; Address of CLUB
     53    10 A4              D0    0252   868          MOVL    CLUB$L_LOCAL_CSB(R4),R3   ; Local system CSB address
   11 1C A4    02              E2    0256   869          BBSS    #CLUB$V_SHUTDOWN, -       ; Set shutdown mode in CLUB
                                     025B   870                  CLUB$L_FLAGS(R4),40$      ;  and branch if already set
   00 60 A3    0A              E2    025B   871          BBSS    #CSB$V_SHUTDOWN, -        ; Set shutdown mode in CSB
                                     0260   872                  CSB$L_STATUS(R3),20$
            FD9D'                30    0260   873 20$:    BSBW    CNX$SCAN_CSBS             ; Iterate over all CSBs
          06 50                  E9    0263   874          BLBC    R0,40$                    ; Branch when done
   00 60 A3    1A              E2    0266   875          BBSS    #CSB$V_SEND_STATUS, -     ; Request that status be sent
                                     026B   876                  CSB$L_STATUS(R3),30$
                                05    026B   877 30$:    RSB
                                     026C   878
            02E6                30    026C   879 40$:    BSBW    SCAN                      ; Look for things to do
         50    01                3C    026F   880          MOVZWL  #SS$_NORMAL,R0            ; Return success status
                                3C    BA    0272   881          POPR    #^M<R2,R3,R4,R5>          ; Restore registers
                                05    0274   882          RSB                               ; Return
```

```
                     0275   884          .SBTTL  CNX$SEND_ALL_STATUS - Send Status to All Flagged Systems
                     0275   885
                     0275   886 ;++
                     0275   887 ;
                     0275   888 ; FUNCTIONAL DESCRIPTION:
                     0275   889 ;
                     0275   890 ;       This routine is called to send our status to all systems
                     0275   891 ;       for whom the send status flag is set.
                     0275   892 ;       The routine returns synchronously, status will eventually
                     0275   893 ;       get sent or the link will break and the status will never
                     0275   894 ;       be sent.
                     0275   895 ;
                     0275   896 ; CALLING SEQUENCE:
                     0275   897 ;
                     0275   898 ;       JSB     CNX$SEND_ALL_STATUS
                     0275   899 ;       IPL is IPL$_SCS
                     0275   900 ;
                     0275   901 ; INPUT PARAMETERS:
                     0275   902 ;
                     0275   903 ;       NONE
                     0275   904 ;
                     0275   905 ; OUTPUT PARAMETERS:
                     0275   906 ;
                     0275   907 ;       R4 is address of CLUB
                     0275   908 ;
                     0275   909 ; COMPLETION CODES:
                     0275   910 ;
                     0275   911 ;       R0
                     0275   912 ;               Success (even): Status will be sent or the link will break
                     0275   913 ;               Failure (odd): Status cannot be sent, probably because storage
                     0275   914 ;                       cannot be allocated
                     0275   915 ;
                     0275   916 ; SIDE EFFECTS:
                     0275   917 ;
                     0275   918 ;       R1 is destroyed
                     0275   919 ;
                     0275   920 ;--
                     0275   921
                     0275   922 CNX$SEND_ALL_STATUS::
            2C   BB  0275   923          PUSHR   #^M<R2,R3,R5>                   ; Save registers
     55     01   D0  0277   924          MOVL    #1,R5                           ; Init summary status
        FD83'   30  027A   925          BSBW    CNX$SCAN_CSBS                   ; Look for CSB needing status
      13 50     E9  027D   926          BLBC    R0, 20$                         ; Branch when done
0D 60 A3  00    E0  0280   927          BBS     #CSB$V_LONG_BREAK, -            ; Ignore nodes with permanently
                    0285   928                  CSB$L_STATUS(R3),10$            ;   connections
08 60 A3  1A    E1  0285   929          BBC     #CSB$V_SEND_STATUS, -           ; Branch if status not requested
                    028A   930                  CSB$L_STATUS(R3),10$
            0D  10  028A   931          BSBB    SEND_STATUS                     ; Queue status message
      03 50     E8  028C   932          BLBS    R0,10$                          ; Branch on success
     55     50   D0  028F   933          MOVL    R0,R5                           ; Save status
            05  0292   934 10$:     RSB
                0293   935
     50     55   D0  0293   936 20$:     MOVL    R5,R0                           ; Fetch status
            2C   BA  0296   937          POPR    #^M<R2,R3,R5>                   ; Restore registers
            05  0298   938          RSB                                     ; Return
```

```
                                .SBTTL  SEND_STATUS - Send Status to a System

                                ;++
                                ;
                                ; FUNCTIONAL DESCRIPTION:
                                ;
                                ;       This routine is called to send our status to a system.
                                ;       The routine returns synchronously, status will eventually
                                ;       get sent or the link will break and the status will never
                                ;       be sent.
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;       JSB     SEND_STATUS
                                ;       IPL is IPL$_SCS
                                ;
                                ; INPUT PARAMETERS:
                                ;
                                ;       R3: CSB of new system
                                ;
                                ; OUTPUT PARAMETERS:
                                ;
                                ;       NONE
                                ;
                                ; COMPLETION CODES:
                                ;
                                ;       R0
                                ;               Success (even): Status will be sent or the link will break
                                ;               Failure (odd): Status cannot be sent, probably because storage
                                ;                       cannot be allocated
                                ;
                                ; SIDE EFFECTS:
                                ;
                                ;       R1 is destroyed
                                ;
                                ;--

                                SEND_STATUS:
                    3C    BB    PUSHR   #^M<R2,R3,R4,R5>        ; Save registers
23 60 A3            18    E0    BBS     #CSB$V_LOCAL, -         ; Skip if local system
                                        CSB$L_STATUS(R3),20$
1E 60 A3            00    E0    BBS     #CSB$V_LONG_BREAK, -    ; Skip if long break
                                        CSB$L_STATUS(R3),20$
                 FD58'    30    BSBW    CNX$ALLOC_CDRP_ONLY
               1B 50      E9    BLBC    R0,30$                  ; No memory available
4C A5            C9'AF    9E    MOVAB   B^BUILD_STS_MSG, -      ; Address of routine to build
                                        CDRP$L_MSGBLD(R5)       ;  status message
00 60 A3            1A    E5    BBCC    #CSB$V_SEND_STATUS, -   ; Clear "send status" request
                                        CSB$L_STATUS(R3),10$
                 FD48'    30    BSBW    CNX$SEND_FORGET         ; Setup immediate return
            55      53    D0    MOVL    R3,R5                   ; Address of CSB
 50       0000'CF    9E    MOVAB   SNDSTS_MSG,R0           ; Sending status message text
                 FD3D'    30    BSBW    CNX$CONFIG_CHANGE       ; Make this state known
            50      01    D0    MOVL    S^#SS$_NORMAL,R0        ; Return success
                    3C    BA    POPR    #^M<R2,R3,R4,R5>        ; Restore registers
                          05    RSB                             ; Return
```

```
                     02C9   998          .SBTTL  BLD_STS_MSG - Build Status Message
                     02C9   999
                     02C9  1000 ;++
                     02C9  1001 ;
                     02C9  1002 ; FUNCTIONAL DESCRIPTION:
                     02C9  1003 ;
                     02C9  1004 ;       This routine builds a status message describing the local node and the
                     02C9  1005 ;       cluster of which it is a member.
                     02C9  1006 ;
                     02C9  1007 ; CALLING SEQUENCE:
                     02C9  1008 ;
                     02C9  1009 ;       JSB     BLD_STS_MSG
                     02C9  1010 ;       IPL is IPL$_SCS
                     02C9  1011 ;
                     02C9  1012 ; INPUT PARAMETERS:
                     02C9  1013 ;
                     02C9  1014 ;       R2:     Address of message buffer
                     02C9  1015 ;       R3:     Address of CSB
                     02C9  1016 ;       R4:     Address of PDT
                     02C9  1017 ;       R5:     Address of CDRP
                     02C9  1018 ;
                     02C9  1019 ; OUTPUT PARAMETERS:
                     02C9  1020 ;
                     02C9  1021 ;       NONE
                     02C9  1022 ;
                     02C9  1023 ; COMPLETION CODES:
                     02C9  1024 ;
                     02C9  1025 ;       NONE
                     02C9  1026 ;
                     02C9  1027 ; SIDE EFFECTS:
                     02C9  1028 ;
                     02C9  1029 ;       R0 and R1 are destroyed
                     02C9  1030 ;
                     02C9  1031 ;--
                     02C9  1032
                     02C9  1033 BUILD_STS_MSG:
   08 A2    01   90  02C9  1034          MOVB    #CLSMSG$K_FAC_CNX, -    ; Facility identifier
                     02CD  1035                  CLSMSG$B_FACILITY(R2)
   09 A2    01   90  02CD  1036          MOVB    #CLMCNX$K_FNC_STATUS, - ; Facility specific function code
                     02D1  1037                  CLSMSG$B_FUNC(R2)
   51    64 A3   D0  02D1  1038          MOVL    CSB$L_CLUB(R3),R1       ; Address of CLUB
      0C A2   94  02D5  1039          CLRB    CLMSTS$B_FLAGS(R2)      ; Clear flags bits
04 1C A1    00   E1  02D8  1040          BBC     #CLUB$V_CLUSTER, -      ; Branch if bit not set
                     02DD  1041                  CLUB$L_FLAGS(R1),10$
   0C A2    01   88  02DD  1042          BISB    #CLMSTS$M_CLUSTER, -    ; Set bit in message
                     02E1  1043                  CLMSTS$B_FLAGS(R2)
04 1C A1    01   E1  02E1  1044 10$:     BBC     #CLUB$V_QF_ACTIVE, -    ; Branch if bit not set
                     02E6  1045                  CLUB$L_FLAGS(R1),20$
   0C A2    02   88  02E6  1046          BISB    #CLMSTS$M_QF_ACTIVE, -  ; Set bit in message
                     02EA  1047                  CLMSTS$B_FLAGS(R2)
04 1C A1    02   E1  02EA  1048 20$:     BBC     #CLUB$V_SHUTDOWN, -     ; Branch if bit not set
                     02EF  1049                  CLUB$L_FLAGS(R1),30$
   0C A2    04   88  02EF  1050          BISB    #CLMSTS$M_SHUTDOWN, -   ; Set bit in message
                     02F3  1051                  CLMSTS$B_FLAGS(R2)
0E A2    20 A1   B0  02F3  1052 30$:     MOVW    CLUB$W_QUORUM(R1), -    ; Cluster quorum
                     02F8  1053                  CLMSTS$W_CQUORUM(R2)
10 A2    22 A1   B0  02F8  1054          MOVW    CLUB$W_VOTES(R1), -     ; Cluster votes
```

```
                                02FD  1055                    CLMSTS$W_CVOTES(R2)
            12 A2    24 A1   B0 02FD  1056          MOVW      CLUB$W_NODES(R1), -     ; Cluster nodes
                             0302  1057                    CLMSTS$W_NODES(R2)
               50    10 A1   D0 0302  1058          MOVL      CLUB$L_LOCAL_CSB(R1),R0 ; Address of local CSB
            14 A2    52 A0   B0 0306  1059          MOVW      CSB$W_QUORUM(R0), -     ; Node quorum
                             030B  1060                    CLMSTS$W_NQUORUM(R2)
            16 A2    50 A0   B0 030B  1061          MOVW      CSB$W_VOTES(R0), -      ; Nodes votes
                             0310  1062                    CLMSTS$W_NVOTES(R2)
            1C A2    2C A1   7D 0310  1063          MOVQ      CLUB$Q_FTIME(R1), -     ; Cluster formation time
                             0315  1064                    CLMSTS$Q_FTIME(R2)
            24 A2    3C A1   7D 0315  1065          MOVQ      CLUB$Q_LST_TIME(R1), -  ; Time stamp of last transaction
                             031A  1066                    CLMSTS$Q_LST_TIME(R2)
            18 A2    56 A0   B0 031A  1067          MOVW      CSB$W_QDVOTES(R0), -    ; Quorum disk votes proposed by local
                             031F  1068                    CLMSTS$W_QDVOTES(R2)    ;   node
            1A A2    54 A0   B0 031F  1069          MOVW      CSB$W_LCKDIRWT(R0), -   ; Lock manager directory system weight
                             0324  1070                    CLMSTS$W_LCKDIRWT(R2)
            2C A2    68 A1   D0 0324  1071          MOVL      CLUB$L_MAX_XTN(R1), -   ; Largest transaction code
                             0329  1072                    CLMSTS$L_MAX_XTN(R2)    ;  seen
            40 A2    74 A0   7D 0329  1073          MOVQ      CSB$Q_REFTIME(R0), -    ; Put local reference time into message
                             032E  1074                    CLMSTS$Q_REFTIME(R2)
                        3C   BB 032E  1075          PUSHR     #^M<R2,R3,R4,R5>        ; Save registers
      30 A2   00B8 C1   10   28 0330  1076          MOVC3     #CLMSTS$S_QDISK, -      ; Quorum disk name
                             0337  1077                    CLUB$T_QDNAME(R1), -    ; from SYSGEN parameter
                             0337  1078                    CLMSTS$B_QDISK(R2)
                        3C   BA 0337  1079          POPR      #^M<R2,R3,R4,R5>        ; Restore registers
                             05   0339  1080          RSB
```

```
                                033A  1082          .SBTTL  CNX$RCVD_STATUS - Status Received from a System
                                033A  1083
                                033A  1084 ;++
                                033A  1085 ;
                                033A  1086 ; FUNCTIONAL DESCRIPTION:
                                033A  1087 ;
                                033A  1088 ;       This routine is called when a status message is received.
                                033A  1089 ;       Data from the message is copied in the CSB.  A cluster
                                033A  1090 ;       transition may be initiated as a result of seeing the
                                033A  1091 ;       message.
                                033A  1092 ;
                                033A  1093 ; CALLING SEQUENCE:
                                033A  1094 ;
                                033A  1095 ;       JSB     CNX$RCVD_STATUS
                                033A  1096 ;       IPL is IPL$_SCS
                                033A  1097 ;
                                033A  1098 ; INPUT PARAMETERS:
                                033A  1099 ;
                                033A  1100 ;       R2:     Message address
                                033A  1101 ;       R3:     CSB of sending system
                                033A  1102 ;       R4:     PDT address
                                033A  1103 ;
                                033A  1104 ; OUTPUT PARAMETERS:
                                033A  1105 ;
                                033A  1106 ;       NONE
                                033A  1107 ;
                                033A  1108 ; COMPLETION CODES:
                                033A  1109 ;
                                033A  1110 ;       NONE
                                033A  1111 ;
                                033A  1112 ; SIDE EFFECTS:
                                033A  1113 ;
                                033A  1114 ;       R0-R5 may be destroyed.
                                033A  1115 ;--
                                033A  1116
                                033A  1117 CNX$RCVD_STATUS::
                          55    53   D0  033A  1118          MOVL    R3,R5                   ; Address of CSB
                    50    0000'CF   9E  033D  1119          MOVAB   RCVSTS_MSG,R0           ; Sending status message text
                        FCBB'   30      0342  1120          BSBW    CNX$CONFIG_CHANGE       ; Make this state known
                          0C    BB      0345  1121          PUSHR   #^M<R2,R3>              ; Save needed registers
                      54    64  A3   D0  0347  1122          MOVL    CSB$L_CLUB(R3),R4       ; Address of CLUB
          60 A3    00000700    8F   CA  034B  1123          BICL2   #<CSB$M_CLUSTER ! -     ; Clear out received status bits in CSB
                                       0353  1124                    CSB$M_QF_ACTIVE ! -
                                       0353  1125                    CSB$M_SHUTDOWN>, -
                                       0353  1126                  CSB$L_STATUS(R3)
                05 0C A2    00   E1  0353  1127          BBC     #CLMSTS$V_CLUSTER, -    ; Branch if bit clear
                                       0358  1128                  CLMSTS$B_FLAGS(R2),1$
                00 60 A3    08   E2  0358  1129          BBSS    #CSB$V_CLUSTER, -       ; Set bit in CSB
                                       035D  1130                  CSB$L_STATUS(R3),1$
                05 0C A2    01   E1  035D  1131 1$:      BBC     #CLMSTS$V_QF_ACTIVE, -  ; Branch if bit clear
                                       0362  1132                  CLMSTS$B_FLAGS(R2),2$
                00 60 A3    09   E2  0362  1133          BBSS    #CSB$V_QF_ACTIVE, -     ; Set bit in CSB
                                       0367  1134                  CSB$L_STATUS(R3),2$
                05 0C A2    02   E1  0367  1135 2$:      BBC     #CLMSTS$V_SHUTDOWN, -   ; Branch if bit clear
                                       036C  1136                  CLMSTS$B_FLAGS(R2),3$
                00 60 A3    0A   E2  036C  1137          BBSS    #CSB$V_SHUTDOWN, -      ; Set bit in CSB
                                       0371  1138                  CSB$L_STATUS(R3),3$
```

```
                00 60 A3    19   E2  0371  1139 3$:     BBSS    #CSB$V_STATUS_RCVD, -   ; Add status received flag
                                     0376  1140                 CSB$L_STATUS(R3),4$
                19 60 A3    01   E0  0376  1141 4$:     BBS     #CSB$V_MEMBER, -        ; Branch if remote node is a member
                                     037B  1142                 CSB$L_STATUS(R3),5$     ;   of the local cluster
                52 A3    14 A2   B0  037B  1143          MOVW    CLMSTS$W_NQUORUM(R2), - ; Remote node's quorum
                                     0380  1144                 CSB$W_QUORUM(R3)
                50 A3    16 A2   B0  0380  1145          MOVW    CLMSTS$W_NVOTES(R2), -  ; Remote node's votes
                                     0385  1146                 CSB$W_VOTES(R3)
                6E A3    12 A2   B0  0385  1147          MOVW    CLMSTS$W_NODES(R2), -   ; Number of nodes in  emote
                                     038A  1148                 CSB$W_NODES(R3)         ;   node's cluster
                56 A3    18 A2   B0  038A  1149          MOVW    CLMSTS$W_QDVOTES(R2), - ; Proposed votes for quorum disk
                                     038F  1150                 CSB$W_QDVOTES(R3)
                54 A3    1A A2   B0  038F  1151          MOVW    CLMSTS$W_LCKDIRWT(R2), - ; Lock manager directory system weight
                                     0394  1152                 CSB$W_LCKDIRWT(R3)
          50    68 A4    2C A2   C3  0394  1153 5$:     SUBL3   CLMSTS$L_MAX_XTN(R2), - ; Update largest transaction seen
                                     039A  1154                 CLUB$L_MAX_XTN(R4),R0
                         05   18  039A  1155          BGEQ    10$                     ; No update needed
                68 A4    2C A2   D0  039C  1156          MOVL    CLMSTS$L_MAX_XTN(R2), - ; Update local information
                                     03A1  1157                 CLUB$L_MAX_XTN(R4)
                 50   00B8 C4    9A  03A1  1158 10$:    MOVZBL  CLUB$T_QDNAME(R4),R0    ; Is local name defined?
                         1D   13  03A6  1159          BEQL    20$                     ; If equal, then can't be the same
                   30 A2   50    91  03A8  1160          CMPB    R0,CLMSTS$B_QDISK(R2)   ; Compare to received name length
                         17   12  03AC  1161          BNEQ    20$                     ; Branch if different
                         0C   BB  03AE  1162          PUSHR   #^M<R2,R3>              ; Save registers
          31 A2   00B9 C4    50  29  03B0  1163          CMPC3   R0,CLUB$T_QDNAME+1(R4), - ; Quorum disk names match?
                                     03B7  1164                 CLMSTS$B_QDISK+1(R2)
                         0C   BA  03B7  1165          POPR    #^M<R2,R3>              ; Restore registers
                         0A   12  03B9  1166          BNEQ    20$                     ; Branch if disk names differ
                05 60 A3    03   E2  03BB  1167          BBSS    #CSB$V_QF_SAME, -       ; Remember that disk names match
                                     03C0  1168                 CSB$L_STATUS(R3),20$    ;   and branch if already known to match
                00 60 A3    1A   E2  03C0  1169          BBSS    #CSB$V_SEND_STATUS, -   ; Just set same bit, send status so
                                     03C5  1170                 CSB$L_STATUS(R3),20$    ;   remote node has same opportunity
                                     03C5  1171 20$:
                05 60 A3    01   E0  03C5  1172          BBS     #CSB$V_MEMBER, -        ; Branch if remote node is a
                                     03CA  1173                 CSB$L_STATUS(R3),30$    ;   member of the local cluster
                74 A3    40 A2   7D  03CA  1174          MOVQ    CLMSTS$Q_REFTIME(R2), - ; Store reference time
                                     03CF  1175                 CSB$Q_REFTIME(R3)
                                     03CF  1176 30$:
                                     03CF  1177 ;
                                     03CF  1178 ; Dispatch on state of this node
                                     03CF  1179 ;
                6E 1C A4    13   E1  03CF  1180          BBC     #CLUB$V_INIT, -         ; Branch to exit if initialization
                                     03D4  1181                 CLUB$L_FLAGS(R4),300$   ;   flag is not yet set
                05 60 A3    01   E1  03D4  1182          BBC     #CSB$V_MEMBER, -        ; Branch if remote node is not
                                     03D9  1183                 CSB$L_STATUS(R3),100$   ; a member of the local cluster
                                     03D9  1184 ;
                                     03D9  1185 ; The local node is part of an existing cluster.
                                     03D9  1186 ; Update quorum calculation to reflect any change seen by the remote node,
                                     03D9  1187 ; for example a change in status of the quorum disk.
                                     03D9  1188 ;
                      FC24'  30      03D9  1189          BSBW    CNX$CHECK_QUORUM        ; Perform dynamic quorum check
                        64   11      03DC  1190          BRB     300$                    ; Branch to common exit
                                     03DE  1191
                4D 1C A4    00   E1  03DE  1192 100$:    BBC     #CLUB$V_CLUSTER, -      ; Branch if local node is not
                                     03E3  1193                 CLUB$L_FLAGS(R4), 200$  ; a cluster member
                                     03E3  1194 ;
                                     03E3  1195 ; Local node is a cluster member.
```

```
                                03E3  1195 ;
                                03E3  1197 ; If remote node is a member of another cluster, bugcheck one of the
                                03E3  1198 ; clusters.  It is illegal for two clusters ever to see each other.
                                03E3  1199 ;
          5A 60 A3    08    E1  03E3  1200         BBC     #CSB$V_CLUSTER, -       ; Branch if remote node is not a
                                03E8  1201                 CSB$L_STATUS(R3), 300$  ; member of another cluster
          20 A2    30 A4    D1  03E8  1202         CMPL    CLUB$Q_FTIME+4(R4), -   ; Compare high order foundation times
                                03ED  1203                 CLMSTS$Q_FTIME+4(R2)
                      35    1A  03ED  1204         BGTRU   110$                    ; We are younger, so we die
                      38    1F  03EF  1205         BLSSU   120$                    ; Other is younger and should die
          1C A2    2C A4    D1  03F1  1206         CMPL    CLUB$Q_FTIME(R4), -     ; Compare low order foundation times
                                03F6  1207                 CLMSTS$Q_FTIME(R2)
                      2C    1A  03F6  1208         BGTRU   110$                    ; We are younger, so we die
                      2F    1F  03F8  1209         BLSSU   120$                    ; Other is younger and should die
          28 A2    40 A4    D1  03FA  1210         CMPL    CLUB$Q_LST_TIME+4(R4), -; Compare high order transition times
                                03FF  1211                 CLMSTS$Q_LST_TIME+4(R2)
                      23    1A  03FF  1212         BGTRU   110$                    ; We are younger, so we die
                      26    1F  0401  1213         BLSSU   120$                    ; Other is younger and should die
          24 A2    3C A4    D1  0403  1214         CMPL    CLUB$Q_LST_TIME(R4), -  ; Compare low order transition times
                                0408  1215                 CLMSTS$Q_LST_TIME(R2)
                      1A    1A  0408  1216         BGTRU   110$                    ; We are younger, so we die
                      1D    1F  040A  1217         BLSSU   120$                    ; Other is younger and should die
                      0C    BB  040C  1218         PUSHR   #^M<R2,R3>              ; Save registers
             52    10 A4    D0  040E  1219         MOVL    CLUB$L_LOCAL_CSB(R4),R2 ; Local CSB address
             52    68 A2    D0  0412  1220         MOVL    CSB$L_SB(R2),R2         ; SB for local system
             53    68 A3    D0  0416  1221         MOVL    CSB$L_SB(R3),R3         ; SB for remote system
       18 A3    18 A2    06    29  041A  1222         CMPC3   #SB$S_SYSTEMID, -       ; Compare system ID's
                                0420  1223                 SB$B_SYSTEMID(R2), -
                                0420  1224                 SB$B_SYSTEMID(R3)
                      0C    BA  0420  1225         POPR    #^M<R2,R3>              ; Restore registers
                      05    1F  0422  1226         BLSSU   120$                    ; Other should die
                                0424  1227 110$:
                                0424  1228 ;
                                0424  1229 ; Should put code in R0 to explain the bugcheck
                                0424  1230 ;
                   FBD9'    30  0424  1231         BSBW    CNX$BUGCHECK_CLUSTER    ; Bring down local cluster
                     19    11  0427  1232         BRB     300$                    ; Exit
                                0429  1233 ;
                                0429  1234 ; Get here if we believe that the other cluster should die.
                                0429  1235 ; Send it status which may help it to realize this.
                                0429  1236 ;
    00 60 A3    1A    E2  0429  1237 120$:  BBSS    #CSB$V_SEND_STATUS, -   ; Request that status be sent
                                042E  1238                 CSB$L_STATUS(R3),130$
                      12    11  042E  1239 130$:  BRB     300$                    ; Branch to common exit
                                0430  1240 ;
                                0430  1241 ; Get here if local node is not part of a cluster
                                0430  1242 ;
          01    58 A4    91  0430  1243 200$:  CMPB    CLUB$B_CUR_CODE(R4), -  ; Is this node forming a cluster?
                                0434  1244                 #CLMCNX$K_XTN_FORM
                      0A    12  0434  1245         BNEQ    210$                    ; Branch if not
    05 0C A2    00    E1  0436  1246         BBC     #CLMSTS$V_CLUSTER, -    ; Branch if remote node is not
                                043B  1247                 CLMSTS$B_FLAGS(R2),210$ ;  a cluster member
    00 1C A4    11    E2  043B  1248         BBSS    #CLUB$V_UNLOCK, -       ; Request failure/unlock to
                                0440  1249                 CLUB$L_FLAGS(R4),210$   ;  abort cluster formation
                      00    11  0440  1250 210$:  BRB     300$                    ; Branch to common exit
                                0442  1251
                                0442  1252 ;
```

```
                     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  1261          .SBTTL  CNX$RCVD_TRNSTS - Received Transition Status Request Message
                        044A  1262
                        044A  1263 ;++
                        044A  1264 ;
                        044A  1265 ; FUNCTIONAL DESCRIPTION:
                        044A  1266 ;
                        044A  1267 ;       This routine is called when a transition status request message
                        044A  1268 ;       is received.  A reply is sent indicating whether or not the specified
                        044A  1269 ;       transition in in or prior to Phase 0, in or post to Phase 2, or in Phase
                        044A  1270 ;       1 while this node does/doesn't have a connection to the coordinator.
                        044A  1271 ;       The purpose of this message is to allow a node caught in Phase 1 when
                        044A  1272 ;       its connection to the coordinator breaks to figure out what to do.
                        044A  1273 ;
                        044A  1274 ; CALLING SEQUENCE:
                        044A  1275 ;
                        044A  1276 ;       JSB     CNX$RCVD_TRNSTS
                        044A  1277 ;       IPL is IPL$_SCS
                        044A  1278 ;
                        044A  1279 ; INPUT PARAMETERS:
                        044A  1280 ;
                        044A  1281 ;       R2:     Message address
                        044A  1282 ;       R3:     CSB of sending system
                        044A  1283 ;       R4:     PDT address
                        044A  1284 ;       R5:     CDRP address (uninitialized)
                        044A  1285 ;
                        044A  1286 ; OUTPUT PARAMETERS:
                        044A  1287 ;
                        044A  1288 ;       NONE
                        044A  1289 ;
                        044A  1290 ; COMPLETION CODES:
                        044A  1291 ;
                        044A  1292 ;       NONE
                        044A  1293 ;
                        044A  1294 ; SIDE EFFECTS:
                        044A  1295 ;
                        044A  1296 ;       R0-R5 may be destroyed.
                        044A  1297 ;--
                        044A  1298
                        044A  1299 CNX$RCVD_TRNSTS::
                  52    DD  044A  1300          PUSHL   R2                      ; Save address of message
              FB81'     30  044C  1301          BSBW    CNX$INIT_CDRP           ; Initialize the CDRP for the response
                04      BA  044F  1302          POPR    #^M<R2>                 ; Restore address of message
           54   64 A3    D0  0451  1303          MOVL    CSB$L_CLUB(R3),R4       ; Address of CLUB
50    34 A4   0C A2    C3  0455  1304          SUBL3   CLMCNX$L_XTN_ID(R2), -  ; Compare transition ID in message with
                            045B  1305                  CLUB$L_LST_XTN(R4),R0   ;   last completed transition
                  4E    14  045B  1306          BGTR    90$                     ; Branch if inconsistent
                  2B    13  045D  1307          BEQL    20$                     ; Branch if same and return Phase 2
                            045F  1308 ;
                            045F  1309 ; Here if ID in message is greater than last completed ID
                            045F  1310 ;
50    48 A4   0C A2    C3  045F  1311          SUBL3   CLMCNX$L_XTN_ID(R2), -  ; Compare transition ID in message with
                            0465  1312                  CLUB$L_COR_XTN(R4),R0   ;   current transition
                  44    19  0465  1313          BLSS    90$                     ; Branch if inconsistent
                  1D    14  0467  1314          BGTR    10$                     ; Branch if same and return Phase 0
                            0469  1315 ;
                            0469  1316 ; Here if ID in message matches current/last unsuccessful ID
                            0469  1317 ;
```

```
                50 8F    59 A4    91  0469  1318          CMPB    CLUB$B_CUR_PHASE(R4), - ; Is phase UNLOCK?
                                       046E  1319                  #CLMCNX$K_UNLOCK
                      16    13       046E  1320          BEQL    10$                     ; Branch if UNLOCK and return Phase 0
                40 8F    59 A4    91  0470  1321          CMPB    CLUB$B_CUR_PHASE(R4), - ; Is phase before or after Phase 1?
                                       0475  1322                  #CLMCNX$K_PH1
                      0F    19       0475  1323          BLSS    10$                     ; Branch if before phase 1 and return Phase
                      11    14       0477  1324          BGTR    20$                     ; Branch if after phase 1 and return Phase 2
                                       0479  1325 ;
                                       0479  1326 ; Here if local system is in phase 1 of a transition
                                       0479  1327 ;
                   51    5C A4    D0  0479  1328          MOVL    CLUB$L_COORD(R4),R1     ; CSB of coordinator
                0C 60 A1    00    E0  047D  1329          BBS     #CSB$V_LONG_BREAK, -    ; Branch if broken and return Phase 1B
                                       0482  1330                  CSB$L_STATUS(R1),30$
                      03    DD       0482  1331          PUSHL   #CLMCNX$K_RP_TRNSTS_PH1 ; Return Phase 1
                      14    11       0484  1332          BRB     40$                     ; Branch to common exit
                                       0486  1333
                      01    DD       0486  1334 10$:     PUSHL   #CLMCNX$K_RP_TRNSTS_PH0 ; Return Phase 0
                      10    11       0488  1335          BRB     40$                     ; Branch to common exit
                                       048A  1336
                      04    DD       048A  1337 20$:     PUSHL   #CLMCNX$K_RP_TRNSTS_PH2 ; Return Phase 2
                      0C    11       048C  1338          BRB     40$                     ; Branch to common exit
                                       048E  1339
                      02    DD       048E  1340 30$:     PUSHL   #CLMCNX$K_RP_TRNSTS_PH1B ; Return Phase 1 (coodinator connection bro
                05 13 A2    00    E1  0490  1341          BBC     #CLMCNX$V_RP_TRNSTS_CMT, - ; Branch if commitment not requested
                                       0495  1342                  CLMCNX$B_REPLY(R2),40$
                00 1C A4    14    E2  0495  1343          BBSS    #CLUB$V_BACKOUT, -      ; Set backout inhibit flag in CLUB
                                       049A  1344                  CLUB$L_FLAGS(R4),40$
                   50    05    9A  049A  1345 40$:     MOVZBL  #CLMCNX$K_FNC_DESC,R0   ; Facility specific function code
                      0E69    30  049D  1346          BSBW    CNX$INIT_STD_RESP       ; Init CDRP for standard response
                36 A5    01    90  04A0  1347          MOVB    #1,CDRP$L_VAL3+2(R5)    ; Store success in ACK field
                37 A5    8E    F6  04A4  1348          CVTLB   (SP)+,CDRP$L_VAL3+3(R5) ; Store reply data
                      FB55'   31  04A8  1349          BRW     CNX$RESP_FORGET         ; Send response message and forget it
                                       04AB  1350
                                       04AB  1351 90$:     BUG_CHECK       CNXMGRERR,FATAL ; Inconsistent state of message and this sys
```

```
                        04AF  1353          .SBTTL  START_TIMEOUT - Start random CLUB-based timeout
                        04AF  1354          .SBTTL  START_LONG_TIMEOUT - Start long random CLUB-based timeout
                        04AF  1355
                        04AF  1356 ;++
                        04AF  1357 ;
                        04AF  1358 ; FUNCTIONAL DESCRIPTION:
                        04AF  1359 ;
                        04AF  1360 ;       If the CLUB fork block is free, start a timeout.
                        04AF  1361 ;       An immediate return is made to the caller.  When the
                        04AF  1362 ;       timout occurs, the thread jumps to SCAN.
                        04AF  1363 ;
                        04AF  1364 ; CALLING SEQUENCE:
                        04AF  1365 ;
                        04AF  1366 ;       JSB     START_TIMEOUT
                        04AF  1367 ;       JSB     START_LONG_TIMEOUT
                        04AF  1368 ;       IPL is IPL$_SCS
                        04AF  1369 ;
                        04AF  1370 ; INPUT PARAMETERS:
                        04AF  1371 ;
                        04AF  1372 ;       NONE
                        04AF  1373 ;
                        04AF  1374 ; OUTPUT PARAMETERS:
                        04AF  1375 ;
                        04AF  1376 ;       NONE
                        04AF  1377 ;
                        04AF  1378 ; COMPLETION CODES:
                        04AF  1379 ;
                        04AF  1380 ;       NONE
                        04AF  1381 ;
                        04AF  1382 ; SIDE EFFECTS:
                        04AF  1383 ;
                        04AF  1384 ;       R0-R5 are destroyed
                        04AF  1385 ;
                        04AF  1386 ;--
                        04AF  1387
                        04AF  1388 START_LONG_TIMEOUT:
 50    4E20 8F    3C    04AF  1389          MOVZWL  #20000,R0               ; 2000...19999 ms timeout
             05    11    04B4  1390          BRB     DO_TIMEOUT
                        04B6  1391
                        04B6  1392 START_TIMEOUT:
 50    1388 8F    3C    04B6  1393          MOVZWL  #5000,R0                ; Range is 500..4999 milli-seconds
                        04BB  1394
                        04BB  1395 DO_TIMEOUT:
             50    DD    04BB  1396          PUSHL   R0                      ; Save timeout range
           FDB5    30    04BD  1397          BSBW    CNX$SEND_ALL_STATUS     ; Send status as necessary
 53    6E    0A    C7    04C0  1398          DIVL3   #10,(SP),R3             ; Make range max/10...max-1
 50    8E    53    C3    04C4  1399          SUBL3   R3,(SP)+,R0             ; Update random number range
           FB35'   30    04C8  1400          BSBW    CNX$RANDOM              ; Generate a random number
       53    50    C0    04CB  1401          ADDL2   R0,R3                   ; Save timeout in R3
15 1C A4    1D    E0    04CE  1402          BBS     #CLUB$V_TRANSITION, -   ; Branch if activity in progress
                        04D3  1403                  CLUB$L_FLAGS(R4),30$
 55    00CC C4    9E    04D3  1404          MOVAB   CLUB$B_FORK_BLOCK(R4),R5 ; Address of transition fork block
08 1C A5    00    E0    04D8  1405          BBS     #CLUBFKB$V_FKB_BUSY, -  ; Branch if fork block in use
                        04DD  1406                  CLUBFKB$L_STATUS(R5),30$
       0084 C4    D5    04DD  1407          TSTL    CLUB$L_TQE(R4)          ; TQE in use?
             05    12    04E1  1408          BNEQ    30$
01 1C A4    13    E0    04E3  1409          BBS     #CLUB$V_INIT, -         ; Branch if initialization complete
```

```
                                04E8  1410                    CLUB$L_FLAGS(R4),40$
                             05 04E8  1411 30$:     RSB
                                04E9  1412
                                04E9  1413 40$:
                                04E9  1414 ;
                                04E9  1415 ; Do timeout, time in milli-seconds in R3
                                04E9  1416 ;
                      51   30   3C  04E9  1417 50$:     MOVZWL  #TQE$K_LENGTH,R1        ; Size of timer queue entry
                00000000'GF   16  04EC  1418           JSB     G^EXE$ALONONPAGED       ; Allocate one
                       FB0B'  30  04F2  1419           BSBW    CNX$RESOURCE_CHECK      ; Watch out for totally exhausted resources
                     3A 50    E9  04F5  1420           BLBC    R0,70$                  ; Can't allocate memory
                  08 A2   51  B0  04F8  1421           MOVW    R1,TQE$W_SIZE(R2)       ; Store size
                  0A A2   0F  90  04FC  1422           MOVB    #DYN$C_TQE,TQE$B_TYPE(R2) ; Store type
                  14 A2   54  D0  0500  1423           MOVL    R4,TQE$L_FR4(R2)        ; Store time delay and CLUB address
                0084 C4   52  D0  0504  1424           MOVL    R2,CLUB$L_TQE(R4)       ; Save TQE address in CLUB
                     55   52  D0  0509  1425           MOVL    R2,R5                   ; Move address of TQE
                  0B A5   01  90  050C  1426           MOVB    #TQE$C_SSSNGL,TQE$B_RQTYPE(R5) ; Store type of timer queue entry
                0C A5   37'AF  9E  0510  1427           MOVAB   B^80$,TQE$L_FPC(R5)     ; Store address of timer fork process
52   00   00002710 8F   53  7A  0515  1428           EMUL    R3,#10*1000,#0,R2       ; Get milli-seconds and cvt to 100 ns. units
          50   00000000'GF  7D  051E  1429           MOVQ    G^EXE$GQ_SYSTIME,R0     ; Get current time
                     50   52  C0  0525  1430           ADDL    R2,R0                   ; Add to current time
                     51   53  D8  0528  1431           ADWC    R3,R1
                00000000'GF  16  052B  1432           JSB     G^EXE$INSTIMQ           ; Insert in timer queue
                             05  0531  1433           RSB                             ; Return
                                 0532  1434
                       FACB'  30  0532  1435 70$:     BSBW    CNX$CLUB_WAIT           ; Wait a second or so in the CLUB
                          B2  11  0535  1436           BRB     50$
                                 0537  1437
                                 0537  1438 ; Come here as a timer fork process to proceed
                                 0537  1439 ; Inputs:
                                 0537  1440 ;       R4      CLUB address
                                 0537  1441 ;       R5      TQE address
                                 0537  1442
                     50   55  D0  0537  1443 80$:     MOVL    R5,R0                   ; Address of timer queue entry
                00000000'GF  16  053A  1444           JSB     G^EXE$DEANONPAGED       ; Deallocate it
                0084 C4   55  D1  0540  1445           CMPL    R5,CLUB$L_TQE(R4)       ; Is this timeout wanted?
                          06  12  0545  1446           BNEQ    90$                     ; Branch if not and drop it
                     0084 C4  D4  0547  1447           CLRL    CLUB$L_TQE(R4)          ; Clear pointer to TQE
                          08  10  054B  1448           BSBB    SCAN                    ; Set up return address, look for work
55   00000000'GF   DE  054D  1449 90$:     MOVAL   G^EXE$AL_TQENOREPT,R5   ; Use non-repeating timer queue entry
                          05  0554  1450           RSB
```

```
                        0555  1452          .SBTTL  SCAN - Look for work to do
                        0555  1453
                        0555  1454 ;++
                        0555  1455 ;
                        0555  1456 ; FUNCTIONAL DESCRIPTION:
                        0555  1457 ;
                        0555  1458 ;       This routine is called whenever it might be appropriate to
                        0555  1459 ;       look for something to do.  These instances include timeouts
                        0555  1460 ;       during initialization and after receiving status from nodes.
                        0555  1461 ;
                        0555  1462 ; CALLING SEQUENCE:
                        0555  1463 ;
                        0555  1464 ;       JSB     SCAN
                        0555  1465 ;       IPL is IPL$_SCS
                        0555  1466 ;
                        0555  1467 ; INPUT PARAMETERS:
                        0555  1468 ;
                        0555  1469 ;       NONE
                        0555  1470 ;
                        0555  1471 ; OUTPUT PARAMETERS:
                        0555  1472 ;
                        0555  1473 ;       NONE
                        0555  1474 ;
                        0555  1475 ; COMPLETION CODES:
                        0555  1476 ;
                        0555  1477 ;       NONE
                        0555  1478 ;
                        0555  1479 ; SIDE EFFECTS:
                        0555  1480 ;
                        0555  1481 ;       R0-R5 may be destroyed.
                        0555  1482 ;--
                        0555  1483
          FD1D   30     0555  1484 SCAN:   BSBW    CNX$SEND_ALL_STATUS     ; Send status as needed
        70 50    E9     0558  1485         BLBC    R0,160$                 ; Exit if couldn't send status
6E 1C A4   13    E1     055B  1486         BBC     #CLUB$V_INIT, -         ; Branch if SYSINIT initialization incomplet
                        0560  1487                 CLUB$L_FLAGS(R4),170$
69 1C A4   1D    E0     0560  1488         BBS     #CLUB$V_TRANSITION, -   ; Branch if activity in progress
                        0565  1489                 CLUB$L_FLAGS(R4),170$   ;   and drop thread
 55   00CC C4   9E     0565  1490         MOVAB   CLUB$B_FORK_BLOCK(R4),R5 ; Address of transition fork block
5F 1C A5   00    E0     056A  1491         BBS     #CLUBFKB$V_FKB_BUSY, -  ; Branch if fork block in use
                        056F  1492                 CLUBFKB$L_STATUS(R5),170$
5B 1C A4   00    E0     056F  1493         BBS     #CLUB$V_CLUSTER, -      ; Branch if this node is cluster member
                        0574  1494                 CLUB$L_FLAGS(R4),200$
                        0574  1495 ;
                        0574  1496 ; Local node is not a member of a cluster.
                        0574  1497 ; Investigate joining or forming a cluster.
                        0574  1498 ;
          FA89'  30     0574  1499         BSBW    CNX$INIT_CSBS           ; Clear out locked, selected bits
            51   7C     0577  1500         CLRQ    R1                      ; R1 is last CSB from a cluster node
                        0579  1501                                         ; R2 is number of cluster nodes seen
          FA84'  30     0579  1502         BSBW    CNX$SCAN_CSBS           ; Iterate over all known CSB's
        20 50    E9     057C  1503         BLBC    R0,140$                 ; Branch when done
1A 60 A3   00    E0     057F  1504         BBS     #CSB$V_LONG_BREAK, -    ; Branch to skip if there has been
                        0584  1505                 CSB$L_STATUS(R3), 130$  ;   a long break
15 60 A3   19    E1     0584  1506         BBC     #CSB$V_STATUS_RCVD, -   ; Branch if status not received
                        0589  1507                 CSB$L_STATUS(R3), 130$
07 60 A3   08    E0     0589  1508         BBS     #CSB$V_CLUSTER, -       ; Branch if remote node is a cluster
```

```
                                058E  1509                    CSB$L_STATUS(R3), 110$  ;  member
      0B 60 A3    11    E3    058E  1510            BBCS    #CSB$V_SELECTED, -      ; Mark node selected
                                0593  1511                    CSB$L_STATUS(R3),130$
                  09    11    0593  1512            BRB     130$                    ; Join common exit
                                0595  1513
                  52    D6    0595  1514 110$:      INCL    R2                      ; Count cluster member node
                  51    D5    0597  1515            TSTL    R1                      ; Seen a cluster node before?
                  00    13    0599  1516            BEQL    120$                    ; Branch if no
                                059B  1517 ;
                                059B  1518 ; Could check that R3 and R1 represent CSB's from the same cluster
                                059B  1519 ;
                        51    53    D0    059B  1520 120$:      MOVL    R3,R1                   ; Update last seen cluster node
                              05    059E  1521 130$:      RSB
                                059F  1522
                  51    D5    059F  1523 140$:      TSTL    R1                      ; Any cluster members seen?
                  1F    12    05A1  1524            BNEQ    150$                    ; Branch if members seen
                                05A3  1525 ;
                                05A3  1526 ; Consider whether it is feasible to try to form a cluster
                                05A3  1527 ;
      23 1C A4    12    E0    05A3  1528            BBS     #CLUB$V_NO_FORM, -      ; Branch if cluster formation is
                                05A8  1529                    CLUB$L_FLAGS(R4),160$
         53   10 A4    D0    05A8  1530            MOVL    CLUB$L_LOCAL_CSB(R4),R3 ; Local CSB address
              50 A3    B5    05AC  1531            TSTW    CSB$W_VOTES(R3)         ; If local node has no votes, don't
                  1A    13    05AF  1532            BEQL    160$                    ;   try to form cluster
                FA4C'   30    05B1  1533            BSBW    CNX$QUORUM_CALC         ; Compute votes, quorum, nodes
              51    50    D1    05B4  1534            CMPL    R0,R1                   ; Is there any possibility of a quorum?
                  12    1F    05B7  1535            BLSSU   160$                    ; No quorum possible, go to common exit
       00C8 C4    03    D3    05B9  1536            BITL    #^B11, -                ; Has a foreign cluster been seen recently?
                                05BE  1537                    CLUB$L_FOREIGN_CLUSTER(R4)
                  0B    12    05BE  1538            BNEQ    160$                    ; Branch if yes and don't try to form a clus
                  4D    11    05C0  1539            BRB     FORM_CLUSTER            ; Try to form a cluster
                                05C2  1540
         6E A1    52    B1    05C2  1541 150$:      CMPW    R2,CSB$W_NODES(R1)      ; Do we perhaps see all members?
                  03    1F    05C6  1542            BLSSU   160$                    ; Nothing to do right now
                02BC    31    05C8  1543            BRW     JOIN_CLUSTER            ; Try to join a cluster
                                05CB  1544
                FEE8    31    05CB  1545 160$:      BRW     START_TIMEOUT           ; Begin a timeout and then rescan
                                05CE  1546
                        05    05CE  1547 170$:      RSB                             ; Exit, dropping thread
                                05CF  1548 ;
                                05CF  1549 ; Here if a cluster member looking for something to do
                                05CF  1550 ;
      03 1C A4    17    E1    05CF  1551 200$:      BBC     #CLUB$V_LOST_CNX, -     ; Branch if no cluster connection is broken
                                05D4  1552                    CLUB$L_FLAGS(R4),220$
                018D    31    05D4  1553            BRW     RECONFIG_CLUSTER        ; Need to do failover
                                05D7  1554
                55    00    D2    05D7  1555 220$:      MCOML   #0,R5                   ; Accumulate shutdown bits
                   FA23'    30    05DA  1556            BSBW    CNX$SCAN_CSBS           ; Iterate over all known CSB's
                 0D 50      E9    05DD  1557            BLBC    R0,240$                 ; Branch when done
      07 60 A3    01    E1    05E0  1558            BBC     #CSB$V_MEMBER, -        ; Branch if not a cluster member
                                05E5  1559                    CSB$L_STATUS(R3), 230$
         50   60 A3    D2    05E5  1560            MCOML   CSB$L_STATUS(R3),R0
              55   50    CA    05E9  1561            BICL2   R0,R5                   ; Accumulate AND of all shutdown bits
                        05    05EC  1562 230$:      RSB
                                05EE  1563
         04 55    0A    E1    05ED  1564 240$:      BBC     #CSB$V_SHUTDOWN,R5,250$ ; Branch if not ready to shutdown
                                05F1  1565            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  1581          .SBTTL  FORM_CLUSTER - Try to form a cluster
                        060F  1582
                        060F  1583 ;++
                        060F  1584 ;
                        060F  1585 ; FUNCTIONAL DESCRIPTION:
                        060F  1586 ;
                        060F  1587 ;       This routine tries to form a new cluster.  It assumes that no
                        060F  1588 ;       cluster is known to this node.
                        060F  1589 ;
                        060F  1590 ; CALLING SEQUENCE:
                        060F  1591 ;
                        060F  1592 ;       JSB     FORM_CLUSTER
                        060F  1593 ;       IPL is IPL$_SCS
                        060F  1594 ;
                        060F  1595 ; INPUT PARAMETERS:
                        060F  1596 ;
                        060F  1597 ;       NONE
                        060F  1598 ;
                        060F  1599 ; OUTPUT PARAMETERS:
                        060F  1600 ;
                        060F  1601 ;       NONE
                        060F  1602 ;
                        060F  1603 ; COMPLETION CODES:
                        060F  1604 ;
                        060F  1605 ;       NONE
                        060F  1606 ;
                        060F  1607 ; SIDE EFFECTS:
                        060F  1608 ;
                        060F  1609 ;       R0-R5 are destroyed
                        060F  1610 ;
                        060F  1611 ;--
                        060F  1612
                        060F  1613 FORM_CLUSTER:
            058F    30  060F  1614          BSBW    INIT_TRANSITION         ; Initialization for becoming coordinator
          50 A3     B5  0612  1615          TSTW    CSB$W_VOTES(R3)         ; If local node has no votes, don't
             6A     13  0615  1616          BEQL    50$                     ;   try to form cluster
       58 A4    01    90  0617  1617          MOVB    #CLMCNX$K_XTN_FORM, -   ; FORM transaction
                        061B  1618                  CLUB$B_CUR_CODE(R4)
    2C A4    50 A4    7D  061B  1619          MOVQ    CLUB$Q_CUR_TIME(R4), -  ; Define foundation time of cluster
                        0620  1620                  CLUB$Q_FTIME(R4)
00000018'GF    06    28  0620  1621          MOVC3   #CLUB$S_FSYSID, -       ; Define founding system ID
          26 A4         0627  1622                  G^SCS$GA_LOCALSB+SB$B_SYSTEMID, -
                        0629  1623                  CLUB$B_FSYSID(R4)
             55     D4  0629  1624          CLRL    R5                      ; No CSB for message
     50  0000'CF    9E  062B  1625          MOVAB   TRYFORM_MSG,R0          ; Trying to form cluster message
            F9CD'   30  0630  1626          BSBW    CNX$CONFIG_CHANGE       ; Make status known
            F9CA'   30  0633  1627          BSBW    CNX$SCAN_CSBS           ; Examine all CSBs
          26 50     E9  0636  1628          BLBC    R0,20$                  ; Branch when done
 14 60 A3    19     E1  0639  1629          BBC     #CSB$V_STATUS_RCVD, -   ; Ignore nodes that haven't sent
                        063E  1630                  CSB$L_STATUS(R3),10$    ;   status
 0F 60 A3    1A     E0  063E  1631          BBS     #CSB$V_SEND_STATUS, -   ; Ignore nodes that need status
                        0643  1632                  CSB$L_STATUS(R3),10$
 0A 60 A3    00     E0  0643  1633          BBS     #CSB$V_LONG_BREAK, -    ; Ignore nodes with whom the connection
                        0648  1634                  CSB$L_STATUS(R3),10$    ;   is broken
            50 A3    B5  0648  1635          TSTW    CSB$W_VOTES(R3)         ; Ignore nodes with no votes
               05    13  064B  1636          BEQL    10$
 00 60 A3    11     E3  064D  1637          BBCS    #CSB$V_SELECTED, -      ; Mark node selected
```

```
                                0652  1638                    CSB$L_STATUS(R3),10$
                     53   DD    0652  1639 10$:     PUSHL    R3                      ; Save context register
 008C C3   20   00   6E   00    2C  0654  1640                MOVC5    #0,(SP),#0, -          ; Zero nodemap
                                0658  1641                    #CSB$S_NODEMAP, -
                                0658  1642                    CSB$B_NODEMAP(R3)
                        08   BA  065C  1643                POPR     #^M<R3>
                        05   05  065E  1644                RSB
                                065F  1645
                                065F  1646 20$:
                                065F  1647 ;
                                065F  1648 ; Selected nodes are the possible population from which to form a
                                065F  1649 ; cluster.
                                065F  1650 ;
                    F99E'   30  065F  1651                BSBW     CNX$QUORUM_CALC        ; Compute votes, quorum, nodes
                 51   50    D1  0662  1652                CMPL     R0,R1                  ; Is there any possibility of a quorum?
                      1A    1F  0665  1653                BLSSU    50$                    ; No quorum possible
                    0ACD    30  0667  1654                BSBW     LOCK_NODES             ; Attempt to lock selected nodes
                                066A  1655 ;
                                066A  1656 ; Have locked all possible nodes.
                                066A  1657 ; Assign temporary CSID's to nodes.
                                066A  1658 ;
                    F993'   30  066A  1659                BSBW     CNX$SCAN_CSBS
                  14 50     E9  066D  1660                BLBC     R0,60$                 ; All done
              0B 60 A3      11  0670  1661                BBC      #CSB$V_SELECTED, -     ; Branch if not selected
                                0675  1662                    CSB$L_STATUS(R3),40$
                    F988'   30  0675  1663                BSBW     CNX$ASSIGN_CSID        ; Create a CSID for the selected node
                 05 50      E8  0678  1664                BLBS     R0,40$                 ; Branch on success
              00 1C A4      11  067B  1665                BBCS     #CLUB$V_UNLOCK, -      ; Request that everything be unlocked
                                0680  1666                    CLUB$L_FLAGS(R4), 40$
                        05   05  0680  1667 40$:     RSB                             ; Resume scan
                                0681  1668
                    0BC2    31  0681  1669 50$:     BRW      UNLOCK_ALL             ; Bail out of transaction
                                0684  1670
                    0BAD    30  0684  1671 60$:     BSBW     CNX$CHECK_UNLOCK       ; Check for unlock request
                                0687  1672 ;
                                0687  1673 ; Prepare to describe all candidate nodes to all other candidate nodes
                                0687  1674 ;
                                0687  1675 FORM_DESCRIBE:
                     5A A4      B4  0687  1676                CLRW     CLUB$W_MSGCNT(R4)      ; Count of responses to wait for
                    F973'   30  068A  1677                BSBW     CNX$SCAN_CSBS          ; Scan all CSBs
                  76 50     E9  068D  1678                BLBC     R0,130$                ; Branch if scan done
              31 60 A3      11  0690  1679                BBC      #CSB$V_SELECTED, -     ; Branch if not selected
                                0695  1680                    CSB$L_STATUS(R3),10$
                 51   10 A4    D0  0695  1681                MOVL     CLUB$L_LOCAL_CSB(R4),R1 ; Local CSB address
                 50   4C A3    3C  0699  1682                MOVZWL   CSB$W_CSID_IDX(R3),R0   ; Subject node CSID index
           00 008C C1   50      E2  069D  1683                BBSS     R0,CSB$B_NODEMAP(R1),5$ ; Mark local node able to see node
              1E 60 A3      00  E0  06A3  1684 5$:      BBS      #CSB$V_LONG_BREAK, -   ; Branch if permanent break in
                                06A8  1685                    CSB$L_STATUS(R3),10$   ;   connection
              19 60 A3      18  E0  06A8  1686                BBS      #CSB$V_LOCAL, -        ; Branch if local CSB
                                06AD  1687                    CSB$L_STATUS(R3),10$
                    F950'   30  06AD  1688                BSBW     CNX$ALLOC_CDRP_ONLY    ; Get a fork block
                    F94D'   30  06B0  1689                BSBW     CNX$RESOURCE_CHECK     ; Watch out for exhausted resources
                 11 50      E9  06B3  1690                BLBC     R0,20$                 ; No memory available
                 48 A5      53  D0  06B6  1691                MOVL     R3,CDRP$L_VAL8(R5)     ; Address of node to talk to
                 44 A5      64  7E  06BA  1692                MOVAQ    CLUB$L_CSBQFL(R4), -   ; Address of CSB listhead
                                06BE  1693                    CDRP$L_VAL7(R5)
                     5A A4      B6  06BE  1694                INCW     CLUB$W_MSGCNT(R4)      ; Increment waiting count
```

```
                07   10  06C1  1695          BSBB    40$                             ; Describe nodes
             F93A'   31  06C3  1696          BRW     CNX$SCAN_CSBS_FORK              ; Fork and continue scan
                     06C6  1697
                 05  06C6  1698 10$:        RSB
                     06C7  1699
             F936'   31  06C7  1700 20$:     BRW     CNX$SCAN_CSBS_RETRY             ; Wait and resume scan
                     06CA  1701
                     06CA  1702 ;
                     06CA  1703 ; Tell a node about all of the others
                     06CA  1704 ;
          52   44 B5   D0  06CA  1705 40$:     MOVL    @CDRP$L_VAL7(R5),R2             ; Next CSB to describe
          44 A5   52   D0  06CE  1706          MOVL    R2,CDRP$L_VAL7(R5)              ;  and save it
                     06D2  1707          ASSUME  CLUB$L_CSBQFL EQ 0
             54   52   D1  06D2  1708          CMPL    R2,R4                           ; At end of list?
                  26   13  06D5  1709          BEQL    125$                            ; Branch if done
       EE 60 A2   11   E1  06D7  1710          BBC     #CSB$V_SELECTED, -              ; Branch if not selected
                     06DC  1711                  CSB$L_STATUS(R2),40$
                086C   30  06DC  1712          BSBW    DESCRIBE_NODE                   ; Tell one node about another
          54   64 A3   D0  06DF  1713          MOVL    CSB$L_CLUB(R3),R4               ; Address of CLUB
               12 50   E9  06E3  1714          BLBC    R0,120$                         ; Error sending message
               0A 51   E9  06E6  1715          BLBC    R1,50$                          ; Branch if node is unknown
          50   4C A2   3C  06E9  1716          MOVZWL  CSB$W_CSID_IDX(R2),R0           ; CSID Index of described system
    00 008C C3   50   E2  06ED  1717          BBSS    R0,CSB$B_NODEMAP(R3), -         ; Mark node visible
                     06F3  1718                  50$
    D2 1C A4   11   E1  06F3  1719 50$:     BBC     #CLUB$V_UNLOCK, -               ; Branch if no unlock request
                     06F8  1720                  CLUB$L_FLAGS(R4),40$
                     06F8  1721 ;
    00 1C A4   11   E2  06F8  1722 120$:    BBSS    #CLUB$V_UNLOCK, -               ; Request that all be unlocked
                     06FD  1723                  CLUB$L_FLAGS(R4),125$
             50   55   D0  06FD  1724 125$:    MOVL    R5,R0                           ; Address of CDRP
      00000000'GF   16   0700  1725          JSB     G^EXE$DEANONPAGED               ; Deallocate CDRP
                     0706  1726
                     0706  1727 ;
                     0706  1728 ; terminate threads -- except for the last one
                     0706  1729 ;
          5A A4   B7  0706  1730 130$:    DECW    CLUB$W_MSGCNT(R4)
             01   19  0709  1731          BLSS    140$                            ; Branch when done
                  05  070B  1732          RSB                                     ; Terminate thread
                     070C  1733
           0B25   30  070C  1734 140$:    BSBW    CNX$CHECK_UNLOCK                ; Check unlock request
                     070F  1735
                     070F  1736 ;
                     070F  1737 ; Get here when all nodes have been described to all others
                     070F  1738 ; Now have told every node about every other node and each node's
                     070F  1739 ; CSB contains a map of nodes that can be seen
                     070F  1740 ;
                     070F  1741 FORM_PROPOSE:
           F8EE'   30  070F  1742          BSBW    CNX$OPT_INIT                    ; Compute optimal cluster containing local n
          23 50   E9  0712  1743          BLBC    R0,270$                         ; Can't perform computation (no pool?)
           F8E8'   30  0715  1744          BSBW    CNX$SCAN_CSBS                   ; Scan all CSBs
          15 50   E9  0718  1745          BLBC    R0,260$                         ; Branch when scan is done
    0F 60 A3   11   E1  071B  1746          BBC     #CSB$V_SELECTED, -              ; Branch if CSB not selected
                     0720  1747                  CSB$L_STATUS(R3),250$
          50   4C A3   3C  0720  1748          MOVZWL  CSB$W_CSID_IDX(R3),R0           ; CSID index of CSB
    05 00EC C4   50   E0  0724  1749          BBS     R0, -                           ; Branch if node is in desired cluster
                     072A  1750                  CLUB$B_NODEMAP(R4),250$ ;
    00 60 A3   11   E5  072A  1751          BBCC    #CSB$V_SELECTED, -              ; Clear select bit
```

```
                                072F  1752                    CSB$L_STATUS(R3),250$
                             05 072F  1753 250$:    RSB                                 ; Advance to next iteration
                                0730  1754
                                0730  1755 ;
                                0730  1756 ; Check for quorum
                                0730  1757 ;
                          F8CD' 30  0730  1758 260$:    BSBW    CNX$QUORUM_CALC         ; Compute quorum, votes, nodes
                       51   50  D1  0733  1759          CMPL    R0,R1                   ; Is there any possibility of a quorum?
                            03  1E  0736  1760          BGEQU   290$                    ; Have a quorum!
                         0B0B  31  0738  1761 270$:    BRW     UNLOCK_ALL              ; Bail out of transition
                               073B  1762
                          F8C2' 30  073B  1763 290$:    BSBW    CNX$DIRVEC_ADJ          ; Create directory system vector
                        F7 50   E9  073E  1764          BLBC    R0,270$                 ; Branch on failure
          0094 C4   00000000'GF 7D  0741  1765          MOVQ    G^EXE$GQ_SYSTIME, -     ; Save current time as new
                                074A  1766                  CLUB$Q_NEWTIME(R4)
              009C C4   0094 C4 7D  074A  1767          MOVQ    CLUB$Q_NEWTIME(R4), -   ; Save as reference time also
                                0751  1768                  CLUB$Q_NEWTIME_REF(R4)
                                0751  1769 ;
                                0751  1770 ; Propose cluster -- Send Phase 1 Messages
                                0751  1771 ;
                      70 A4   07  9A  0751  1772          MOVZBL  #CLMCNX$K_FNC_FORM, -   ; Facility specific message code
                                0755  1773                  CLUB$L_CTX0(R4)
                   78 A4   0BFA'CF  9E  0755  1774          MOVAB   W^BLD_FORM_MSG, -       ; Address of routine to build
                                075B  1775                  CLUB$L_CTXT(R4)             ;  status message
                         0901   30  075B  1776          BSBW    SEND_PH1                ; Send phase 1 messages
                                075E  1777 ;
                                075E  1778 ; All nodes have seen Phase 1 and ACKed it.
                                075E  1779 ; Now is the time to send Phase 2.
                                075E  1780 ;
                         095D   31  075E  1781          BRW     SEND_PH2
                                0761  1782 ;
                                0761  1783 ; At the completion of sending Phase 2 messages, this location
                                0761  1784 ; is executed as a thread.
                                0761  1785 ;
                                0761  1786 FORM_FINISH:
                         0322   31  0761  1787          BRW     FORM_JOIN_FINISH        ; Branch to common form/join phase c process
```

```
                     0764  1789          .SBTTL  RECONFIG_CLUSTER - Reconfigure a cluster following a node failure
                     0764  1790
                     0764  1791 ;++
                     0764  1792 ;
                     0764  1793 ; FUNCTIONAL DESCRIPTION:
                     0764  1794 ;
                     0764  1795 ;       This routine tries to reconfigure a cluster following a connection failure.
                     0764  1796 ;       This implements a fast and simple scheme that works only if a failure involv
                     0764  1797 ;       all connections to a failed set of nodes.  A second scheme is necessary to
                     0764  1798 ;       deal with the more complicated case of random connection failures.
                     0764  1799 ;
                     0764  1800 ; CALLING SEQUENCE:
                     0764  1801 ;
                     0764  1802 ;       JSB     RECONFIG_CLUSTER
                     0764  1803 ;       IPL is IPL$_SCS
                     0764  1804 ;
                     0764  1805 ; INPUT PARAMETERS:
                     0764  1806 ;
                     0764  1807 ;       NONE
                     0764  1808 ;
                     0764  1809 ; OUTPUT PARAMETERS:
                     0764  1810 ;
                     0764  1811 ;       NONE
                     0764  1812 ;
                     0764  1813 ; COMPLETION CODES:
                     0764  1814 ;
                     0764  1815 ;       NONE
                     0764  1816 ;
                     0764  1817 ; SIDE EFFECTS:
                     0764  1818 ;
                     0764  1819 ;       R0-R5 are destroyed
                     0764  1820 ;
                     0764  1821 ;--
                     0764  1822
                     0764  1823 RECONFIG_CLUSTER:
           043A   30 0764  1824          BSBW    INIT_TRANSITION         ; Initialization for becoming coordinator
     58 A4   03   90 0767  1825          MOVB    #CLMCNX$K_XTN_RECONFIG, - ; RECONFIGure transaction
                     076B  1826                  CLUB$B_CUR_CODE(R4)
             55   D4 076B  1827          CLRL    R5                      ; No CSB address
      50  0000'CF   9E 076D  1828          MOVAB   RECONFIG_MSG,R0         ; Initiating cluster reconfiguration message
         F888'   30 0772  1829          BSBW    CNX$CONFIG_CHANGE       ; Make this state known
         F888'   30 0775  1830          BSBW    CNX$SCAN_CSBS
        10 50    E9 0778  1831          BLBC    R0,30$                  ; Branch when done
0A 60 A3   01    E1 077B  1832          BBC     #CSB$V_MEMBER, -        ; Ignore nodes that are not members of
                     0780  1833                  CSB$L_STATUS(R3),10$    ;   the local cluster
05 60 A3   00    E0 0780  1834          BBS     #CSB$V_LONG_BREAK, -    ; Ignore nodes with whom the connection
                     0785  1835                  CSB$L_STATUS(R3),10$    ;   is irrevocably broken
00 60 A3   11    E3 0785  1836          BBCS    #CSB$V_SELECTED, -      ; Mark node selected
                     078A  1837                  CSB$L_STATUS(R3),10$
             05   05 078A  1838 10$:    RSB
                     078B  1839
                     078B  1840 30$:
                     078B  1841 ;
                     078B  1842 ; Selected nodes are those proposed for the members of the reconfigured
                     078B  1843 ; cluster.
                     078B  1844 ;
           09A9   30 078B  1845          BSBW    LOCK_NODES              ; Attempt to lock selected nodes
```

```
                          078E  1846
                          078E  1847 RECONFIG_LOCKED:
                          078E  1848 ;
                          078E  1849 ; Now, exchange topology messages with all member nodes
                          078E  1850 ;
          5A A4    B4     078E  1851          CLRW    CLUB$W_MSGCNT(R4)       ; Initialize waiting count
      59 A4   30   90     0791  1852          MOVB    #CLMCNX$K_DATA, -       ; Mark this as Phase 1
                          0795  1853                  CLUB$B_CUR_PHASE(R4)
           F868'  30      0795  1854          BSBW    CNX$SCAN_CSBS           ; Iterate over all CSBs
         44 50    E9      0798  1855          BLBC    R0,90$                  ; Branch when done
  1B 60 A3   11    E1     079B  1856          BBC     #CSB$V_SELECTED, -      ; Branch if not selected
                          07A0  1857                  CSB$L_STATUS(R3),40$
  16 60 A3   18    E0     07A0  1858          BBS     #CSB$V_LOCAL, -         ; Branch if local CSB
                          07A5  1859                  CSB$L_STATUS(R3),40$
  11 60 A3   00    E0     07A5  1860          BBS     #CSB$V_LONG_BREAK, -    ; Branch if connection is
                          07AA  1861                  CSB$L_STATUS(R3),40$    ;   permanently broken
           F853'  30      07AA  1862          BSBW    CNX$ALLOC_WARMCDRP_CSB
           F850'  30      07AD  1863          BSBW    CNX$RESOURCE_CHECK      ; Watch out for exhausted resources
         09 50    E9      07B0  1864          BLBC    R0,50$                  ; No memory available
         5A A4    B6      07B3  1865          INCW    CLUB$W_MSGCNT(R4)       ; Include in wait count
            07    10      07B6  1866          BSBB    60$                     ; Describe nodes
           F845'  31      07B8  1867          BRW     CNX$SCAN_CSBS_FORK      ; Fork and continue scan
                          07BB  1868
                    05    07BB  1869 40$:     RSB                             ; Return and continue scan
                          07BC  1870
           F841'  31      07BC  1871 50$:     BRW     CNX$SCAN_CSBS_RETRY     ; Delay and resume scan
                          07BF  1872
                          07BF  1873 ;
                          07BF  1874 ; Send topology message to one node
                          07BF  1875 ;
          50   0F   9A     07BF  1876 60$:     MOVZBL  #CLMCNX$K_FNC_TOPOLOGY,R0 ; Message code
             0B4D  30      07C2  1877          BSBW    INIT_STD_MSG            ; Standard CDRP message initialization
      4C A5  1022'CF   9E  07C5  1878          MOVAB   W^BLD_TOPOLOGY_MSG, -   ; Address of routine to build
                          07CB  1879                  CDRP$L_MSGBLD(R5)       ;  topology message
           F832'  30      07CB  1880          BSBW    CNX$SEND_MSG_CSB        ; Send message
                          07CE  1881 ;
                          07CE  1882 ;        We are resumed here when the response message arrives.
                          07CE  1883 ;        Registers contain:
                          07CE  1884 ;                R0:     Status
                          07CE  1885 ;                R2:     Address of message buffer
                          07CE  1886 ;                R3:     Address of CSB
                          07CE  1887 ;                R4:     Address of PDT
                          07CE  1888 ;                R5:     Address of CDRP
                          07CE  1889 ;
        0B 50    E9       07CE  1890          BLBC    R0,80$                  ; Branch on no data received
           3D    BB       07D1  1891          PUSHR   #^M<R0,R2,R3,R4,R5>     ; Save registers
008C C3   14 A2   20      07D3  1892          MOVC3   #CSB$S_NODEMAP, -       ; Fill in connectivity in sending node's
                          07DA  1893                  CLMTOP$B_NODEMAP(R2), - ;   CSB
                          07DA  1894                  CSB$B_NODEMAP(R3)
           3D    BA       07DA  1895          POPR    #^M<R0,R2,R3,R4,R5>     ; Restore registers
         0B07    30       07DC  1896 80$:     BSBW    CNX$PROCESS_RESPONSE    ; Deallocate storage
                          07DF  1897 ;
                          07DF  1898 ; terminate threads -- except for the last one
                          07DF  1899 ;
          5A A4    B7     07DF  1900 90$:     DECW    CLUB$W_MSGCNT(R4)
             01    19      07E2  1901          BLSS    100$                    ; Branch when done
                    05    07E4  1902          RSB                             ; Terminate thread
```

```
                                07E5  1903
                         0A4C   30  07E5  1904 100$:   BSBW    CNX$CHECK_UNLOCK         ; Handle a pending unlock request
                                07E8  1905
                                07E8  1906 RECONFIG_PROPOSE:
                                07E8  1907 ;
                                07E8  1908 ; Get here when all nodes in the proposed new cluster have been locked.
                                07E8  1909 ; Compute and send Phase 1 proposal messages.
                                07E8  1910 ;
                          F815' 30  07E8  1911         BSBW    CNX$OPT_INIT             ; Compute optimal subcluster
                        21 50   E9  07EB  1912         BLBC    R0,25$                   ; Can't compute (no pool?)
                          F80F' 30  07EE  1913         BSBW    CNX$SCAN_CSBS            ; Scan all CSB's
                        15 50   E9  07F1  1914         BLBC    R0,20$                   ; Branch when done
                     0F 60 A3   11  E1  07F4  1915         BBC     #CSB$V_SELECTED, -       ; Branch if not selected
                                07F9  1916                 CSB$L_STATUS(R3),10$
                           50   4C A3  3C  07F9  1917         MOVZWL  CSB$W_CSID_IDX(R3),R0    ; CSID index of CSB
                     05 00EC C4   50  E0  07FD  1918         BBS     R0, -                    ; Node is in computed subcluster
                                0803  1919                 CLUB$B_NODEMAP(R4),10$   ;
                      00 60 A3   11  E5  0803  1920         BBCC    #CSB$V_SELECTED, -       ; De-select node not in computed cluster
                                0808  1921                 CSB$L_STATUS(R3),10$
                                05  0808  1922 10$:     RSB                              ; Return and resume scan
                                0809  1923
                                0809  1924 20$:
                          F7F4' 30  0809  1925         BSBW    CNX$DIRVEC_ADJ           ; Adjust size of lock manager directory vect
                        03 50   E8  080C  1926         BLBS    R0,30$                   ; Branch on success
                          0A34  31  080F  1927 25$:     BRW     UNLOCK_ALL
                                0812  1928
                          F7EB' 30  0812  1929 30$:     BSBW    CNX$QUORUM_CALC          ; Calculate and store quorum related paramet
                                0815  1930 ;
                                0815  1931 ; Propose cluster -- Send Phase 1 Messages
                                0815  1932 ;
                           70 A4   08   9A  0815  1933         MOVZBL  #CLMCNX$K_FNC_RECONFIG, - ; Message code
                                0819  1934                 CLUB$L_CTX0(R4)
                     78 A4   0BFA'CF   9E  0819  1935         MOVAB   W^BLD_RECONFIG_MSG, -    ; Address of routine to build
                                081F  1936                 CLUB$L_CTX1(R4)          ;  status message
                                083D  1937         BSBW    SEND_PH1                 ; Send phase 1 messages
                                0822  1938 ;
                                0822  1939 ; All nodes have seen Phase 1 and ACKed it.
                                0822  1940 ; Now is the time to send Phase 2.
                                0822  1941 ;
                                0899  1942         BRW     SEND_PH2
                                0825  1943 ;
                                0825  1944 ; At the completion of sending Phase 2 messages, this location
                                0825  1945 ; is executed as a thread.
                                0825  1946 ;
                                0825  1947 RECONFIG_FINISH:
                          F7D8' 30  0825  1948         BSBW    CNX$SCAN_CSBS            ; Iterate over all CSBs
                        18 50   E9  0828  1949         BLBC    R0,30$                   ; Branch when scan is done
                     12 60 A3   11  E1  082B  1950         BBC     #CSB$V_SELECTED, -       ; Branch if node not selected
                                0830  1951                 CSB$L_STATUS(R3),20$
                           50   1F  D0  0830  1952         MOVL    #CSB$S_NODEMAP-1,R0      ; Byte count of nodemap
                     52   00EC C440  92  0833  1953 10$:     MCOMB   CLUB$B_NODEMAP(R4)[R0],R2
                     008C C340   52  8A  0839  1954         BICB2   R2,CSB$B_NODEMAP(R3)[R0] ; CSB = CSB & CLUB
                           F1 50   F4  083F  1955         SOBGEQ  R0,10$
                                05  0842  1956 20$:     RSB
                                0843  1957
                          0B1D  30  0843  1958 30$:     BSBW    UPDATE_QUORUM            ; Update quorum, votes, nodes
                                0846  1959 ;
```

```
                          0846  1960 ; Handle bookkeeping for removed nodes.
                          0846  1961 ;
          F7B7'  30       0846  1962          BSBW    CNX$SCAN_CSBS           ; Iterate over all CSBs
        13 50    E9       0849  1963          BLBC    R0,60$                  ; Branch when done
0A 60 A3   11    E0       084C  1964          BBS     #CSB$V_SELECTED, -      ; Branch if CSB selected -
                          0851  1965                  CSB$L_STATUS(R3),40$    ;   still a cluster member
08 60 A3   01    E1       0851  1966          BBC     #CSB$V_MEMBER, -        ; Branch if not a cluster member
                          0856  1967                  CSB$L_STATUS(R3),50$
          0BA0   30       0856  1968          BSBW    REMOVE_NODE             ; Mark the node out of the cluster
            03   11       0859  1969          BRB     50$                     ; Branch to common exit
                          085B  1970
          F7A2'  30       085B  1971 40$:     BSBW    CNX$MARK_UNLOCKED       ; Mark node unlocked
                 05       085E  1972 50$:     RSB                             ; Continue scan
                          085F  1973
                          085F  1974 60$:
                          085F  1975 ;
                          085F  1976 ; Recompute value of CLUB$V_LOST_CNX
                          085F  1977 ;
00 1C A4   17    E5       085F  1978          BBCC    #CLUB$V_LOST_CNX, -     ; Clear bit prior to recomputing its
                          0864  1979                  CLUB$L_FLAGS(R4),70$
          F799'  30       0864  1980 70$:     BSBW    CNX$SCAN_CSBS           ; Iterate over CSBs
        10 50    E9       0867  1981          BLBC    R0,220$                 ; Branch when scan is done
0A 60 A3   01    E1       086A  1982          BBC     #CSB$V_MEMBER, -        ; Branch if node is not a cluster member
                          086F  1983                  CSB$L_STATUS(R3), 210$
05 60 A3   00    E1       086F  1984          BBC     #CSB$V_LONG_BREAK, -    ; Branch if no long break has happened
                          0874  1985                  CSB$L_STATUS(R3),210$
00 1C A4   17    E2       0874  1986          BBSS    #CLUB$V_LOST_CNX, -     ; Mark cluster connection lost
                          0879  1987                  CLUB$L_FLAGS(R4),210$
                 05       0879  1988 210$:     RSB
                          087A  1989
     F783'  30            087A  1990 220$:     BSBW    CNX$DIRVEC_FILL         ; Recompute lock manager directory vector
00AC C4     B6            087D  1991          INCW    CLUB$W_MEMSEQ(R4)       ; Advance membership state sequence number
     F77C'  30            0881  1992          BSBW    CNX$MEMBERSHIP_CHANGE   ; Begin table processing to remove nodes
     09BF   31            0884  1993          BRW     UNLOCK_ALL              ; Make sure everything is unlocked
```

```
                     0887  1995          .SBTTL  JOIN_CLUSTER - Try to join a cluster
                     0887  1996
                     0887  1997 ;++
                     0887  1998 ;
                     0887  1999 ; FUNCTIONAL DESCRIPTION:
                     0887  2000 ;
                     0887  2001 ;       This routine is called by a newly booted node that decides to
                     0887  2002 ;       request membership in a cluster.
                     0887  2003 ;
                     0887  2004 ; CALLING SEQUENCE:
                     0887  2005 ;
                     0887  2006 ;       JSB     JOIN_CLUSTER
                     0887  2007 ;       IPL is IPL$_SCS
                     0887  2008 ;
                     0887  2009 ; INPUT PARAMETERS:
                     0887  2010 ;
                     0887  2011 ;       NONE
                     0887  2012 ;
                     0887  2013 ; OUTPUT PARAMETERS:
                     0887  2014 ;
                     0887  2015 ;       NONE
                     0887  2016 ;
                     0887  2017 ; COMPLETION CODES:
                     0887  2018 ;
                     0887  2019 ;       NONE
                     0887  2020 ;
                     0887  2021 ; SIDE EFFECTS:
                     0887  2022 ;
                     0887  2023 ;       R0-R5 are destroyed
                     0887  2024 ;
                     0887  2025 ;--
                     0887  2026
                     0887  2027 JOIN_CLUSTER:
          55   D4   0887  2028          CLRL    R5                              ; Will hold CSB address on return
        F774'  30   0889  2029          BSBW    CNX$SCAN_CSBS                   ; Examine all CSBs
      16 50    E9   088C  2030          BLBC    R0,20$                          ; Branch when done
10 60 A3   19    E1   088F  2031          BBC     #CSB$V_STATUS_RCVD, -           ; Ignore nodes that haven't sent
                     0894  2032                  CSB$L_STATUS(R3),10$            ;   status
0B 60 A3   08    E1   0894  2033          BBC     #CSB$V_CLUSTER, -               ; Ignore non-cluster members
                     0899  2034                  CSB$L_STATUS(R3),10$
06 60 A3   00    E0   0899  2035          BBS     #CSB$V_LONG_BREAK, -            ; Ignore nodes with whom the connection
                     089E  2036                  CSB$L_STATUS(R3),10$            ;   is broken
     55   53   D0   089E  2037          MOVL    R3,R5
        F75C' 31   08A1  2038          BRW     CNX$SCAN_CSBS_EXIT              ; Return CSB address of cluster member
                     08A4  2039
          05   08A4  2040 10$:     RSB
                     08A5  2041
     53   55   D0   08A5  2042 20$:     MOVL    R5,R3                           ; CSB address of node to apply to
          2C   13   08A8  2043          BEQL    50$                             ; No node to apply to at this time
27 60 A3   0A    E0   08AA  2044          BBS     #CSB$V_SHUTDOWN, -              ; Branch if node in shutdown mode
                     08AF  2045                  CSB$L_STATUS(R3),50$
22 60 A3   1A    E0   08AF  2046          BBS     #CSB$V_SEND_STATUS, -           ; Branch if status not yet queued
                     08B4  2047                  CSB$L_STATUS(R3),50$            ;   for transmission
        F749'  30   08B4  2048          BSBW    CNX$ALLOC_CDRP_ONLY             ; Get a CDRP
      1C 50    E9   08B7  2049          BLBC    R0,50$                          ; No memory available
   54   64 A3   D0   08BA  2050          MOVL    CSB$L_CLUB(R3),R4               ; CLUB address
   58 A4   02   90   08BE  2051          MOVB    #CLM_CNX$K_XTN_JOIN, -          ; Transition identifier
```

```
                                B 1
                     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  2061          .SBTTL  CNX$RCVD_ENTER - Received a cluster membership request
                        08D9  2062
                        08D9  2063 ;++
                        08D9  2064 ;
                        08D9  2065 ; FUNCTIONAL DESCRIPTION:
                        08D9  2066 ;
                        08D9  2067 ;       This routine is called when a cluster membership request is received.
                        08D9  2068 ;       If the request looks valid, the protocol for adding a node to a cluster
                        08D9  2069 ;       is executed.
                        08D9  2070 ;
                        08D9  2071 ; CALLING SEQUENCE:
                        08D9  2072 ;
                        08D9  2073 ;       JSB     CNX$RCVD_ENTER
                        08D9  2074 ;       IPL is IPL$_SCS
                        08D9  2075 ;
                        08D9  2076 ; INPUT PARAMETERS:
                        08D9  2077 ;
                        08D9  2078 ;       R2:     Message address
                        08D9  2079 ;       R3:     CSB of sending system
                        08D9  2080 ;       R4:     PDT address
                        08D9  2081 ;
                        08D9  2082 ; OUTPUT PARAMETERS:
                        08D9  2083 ;
                        08D9  2084 ;       NONE
                        08D9  2085 ;
                        08D9  2086 ; COMPLETION CODES:
                        08D9  2087 ;
                        08D9  2088 ;       NONE
                        08D9  2089 ;
                        08D9  2090 ; SIDE EFFECTS:
                        08D9  2091 ;
                        08D9  2092 ;       R0-R5 are destroyed
                        08D9  2093 ;
                        08D9  2094 ;--
                        08D9  2095
                        08D9  2096 CNX$RCVD_ENTER::
            F724'   30  08D9  2097          BSBW    CNX$DEALL_MSG_BUF_CSB   ; Deallocate the message buffer
        55   53     D0  08DC  2098          MOVL    R3,R5                   ; Address of CSB
    50  0000'CF     9E  08DF  2099          MOVAB   MEMREQ_MSG,R0           ; Address of message block
            F719'   30  08E4  2100          BSBW    CNX$CONFIG_CHANGE       ; Received cluster membership request
      54    64 A3     D0  08E7  2101          MOVL    CSB$L_CLUB(R3),R4       ; Address of CLUB
1C A4   20800004 8F     D3  08EB  2102          BITL    #< -                    ; Check for:
                            08F3  2103              CLUB$M_TRANSITION ! -   ;   Transition in progress
                            08F3  2104              CLUB$M_LOST_CNX ! -     ;   Node addition inhibited
                            08F3  2105              CLUB$M_SHUTDOWN>, -     ;   Shutdown in progress
                            08F3  2106          CLUB$L_FLAGS(R4)
              30    12      08F3  2107          BNEQ    20$                     ; Branch to ignore request
        2B 1C A4    00    E1  08F5  2108          BBC     #CLUB$V_CLUSTER, -      ; Branch if this node is not a cluster
                            08FA  2109          CLUB$L_FLAGS(R4), 20$   ;   member
60 A3   04000007 8F     D3  08FA  2110          BITL    #< -                    ; Check for:
                            0902  2111              CSB$M_SEND_STATUS ! -   ;   Requestor is out of date
                            0902  2112              CSB$M_MEMBER ! -        ;   Requestor already a cluster member
                            0902  2113              CSB$M_LONG_BREAK ! -    ;   Connection problem to requestor *** why?
                            0902  2114              CSB$M_REMOVED>, -       ;   Requestor removed from cluster
                            0902  2115          CSB$L_STATUS(R3)
              21    12      0902  2116          BNEQ    20$                     ; Branch to ignore request
        1C 60 A3    19    E1  0904  2117          BBC     #CSB$V_STATUS_RCVD, -   ; Branch if we don't have info on
```

```
                     0909  2118                    CSB$L_STATUS(R3), 20$   ;   the requestor
    70 A4    53  D0  0909  2119            MOVL    R3,CLUB$L_CTX0(R4)      ; Save joining node CSB address
                     090D  2120 ;
                     090D  2121 ; The following piece of code assumes that CNX$SCAN_CSBS returns CSBs in order of
                     090D  2122 ; increasing system ID.
                     090D  2123 ;
       55    01  D0  090D  2124            MOVL    #1,R5                   ; Assume local node has lowest System ID
          F6ED'  30  0910  2125            BSBW    CNX$SCAN_CSBS           ; Iterate over all CSBs
        10 50    E9  0913  2126            BLBC    R0,30$                  ; Must have found something
0A 60 A3   01    E1  0916  2127            BBC     #CSB$V_MEMBER, -        ; Branch if CSB is not for a member
                     091B  2128                    CSB$L_STATUS(R3),20$
02 60 A3   18    E0  091B  2129            BBS     #CSB$V_LOCAL, -         ; Branch if not local CSB
                     0920  2130                    CSB$L_STATUS(R3),10$
            55   D4  0920  2131            CLRL    R5                      ; Local node is not lowest system ID
          F6DB'  31  0922  2132 10$:        BRW     CNX$SCAN_CSBS_EXIT
                     0925  2133
                05   05  0925  2134 20$:        RSB
                     0926  2135
          FC 55    E9  0926  2136 30$:        BLBC    R5,20$                  ; If this is not the right node, drop thread
                     0929  2137 ;
                     0929  2138 ; Here, the transition to add the node begins
                     0929  2139 ;
    58 A4    02  90  0929  2140            MOVB    #CLMCNX$K_XTN_JOIN, -   ; JOIN transaction
                     092D  2141                    CLUB$B_CUR_CODE(R4)
          0271   30  092D  2142            BSBW    INIT_TRANSITION         ; Initialization for becoming coordinator
    55    70 A4  D0  0930  2143            MOVL    CLUB$L_CTX0(R4),R5      ; CSB of joining node
 50    0000'CF   9E  0934  2144            MOVAB   JOIN_MSG,R0             ; Initiating addition of system to cluster
          F6C4'  30  0939  2145            BSBW    CNX$CONFIG_CHANGE       ; Make this state known
          F6C1'  30  093C  2146            BSBW    CNX$SCAN_CSBS           ; Iterate over all CSBs
        0B 50    E9  093F  2147            BLBC    R0, 60$                 ; Branch when done
05 60 A3   01    E1  0942  2148            BBC     #CSB$V_MEMBER, -        ; Branch if not a cluster member
                     0947  2149                    CSB$L_STATUS(R3), 50$
00 60 A3   11    E2  0947  2150            BBSS    #CSB$V_SELECTED, -      ; Select member nodes
                     094C  2151                    CSB$L_STATUS(R3), 50$
                05   05  094C  2152 50$:        RSB
                     094D  2153
       53    70 A4  D0  094D  2154 60$:        MOVL    CLUB$L_CTX0(R4),R3      ; CSB address of joining node
60 A3    00020000 8F  C8  0951  2155            BISL2   #CSB$M_SELECTED, -      ; Mark joining node selected
                     0959  2156                    CSB$L_STATUS(R3)
             F6A4'  30  0959  2157            BSBW    CNX$QUORUM_CALC         ; Calculate quorum related parameters
          51   50  D1  095C  2158            CMPL    R0,R1                   ; Votes >= quorum?
               0A  1E  095F  2159            BGEQU   80$                     ; Branch if yes
    20 A4    22 A4  B1  0961  2160            CMPW    CLUB$W_VOTES(R4), -     ; Is there currently a quorum present?
                     0966  2161                    CLUB$W_QUORUM(R4)
               03  1F  0966  2162            BLSSU   80$                     ; Branch if yes, don't undo quorum by adding
             08DB  31  0968  2163 70$:        BRW     UNLOCK_ALL              ; Abort transition
                     096B  2164
             F692'  30  096B  2165 80$:        BSBW    CNX$ASSIGN_CSID         ; Assign a CSID
            F7 50   E9  096E  2166            BLBC    R0,70$                  ; Branch on failure
                     0971  2167
             07C3   30  0971  2168            BSBW    LOCK_NODES              ; Attempt to lock selected nodes
                     0974  2169
                     0974  2170 JOIN_LOCKED:
                     0974  2171
54   00000000'GF  D0  0974  2172            MOVL    G^CLU$GL_CLUB,R4        ; Address of CLUB
            5A A4  B4  097B  2173            CLRW    CLUB$W_MSGCNT(R4)       ; Count of responses to wait for
                     097E  2174 ;
```

```
                                097E  2175 ; Describe all member nodes to joining system
                                097E  2176 ;
                     F67F'  30  097E  2177          BSBW    CNX$SCAN_CSBS           ; Iterate over all CSBs
                   2D 50    E9  0981  2178          BLBC    R0,30$                  ; Branch when all systems described
          21 60 A3    11    E1  0984  2179          BBC     #CSB$V_SELECTED, -      ; Branch if node not selected
                                0989  2180                  CSB$L_STATUS(R3),10$
          1D 1C A4    11    E0  0989  2181          BBS     #CLUB$V_UNLOCK, -       ; Branch if unlock requested
                                098E  2182                  CLUB$L_FLAGS(R4),15$    ;   and abandon this transition
                     F66F'  30  098E  2183          BSBW    CNX$ALLOC_CDRP_ONLY     ; Get a fork block
                     F66C'  30  0991  2184          BSBW    CNX$RESOURCE_CHECK      ; Watch out for exhausted resources
                   17 50    E9  0994  2185          BLBC    R0,20$                  ; No memory available
                        53  DD  0997  2186          PUSHL   R3                      ; Save scan context
                52      53  D0  0999  2187          MOVL    R3,R2                   ; CSB of node to describe
             53    70 A4    D0  099C  2188          MOVL    CLUB$L_CTX0(R4),R3      ; Address of joining node CSB
                   5A A4    B6  09A0  2189          INCW    CLUB$W_MSGCNT(R4)       ; Increment waiting count
                      3A    10  09A3  2190          BSBB    60$                     ; Provide immediate return address
                      08    BA  09A5  2191          POPR    #^M<R3>                 ; Restore scanning context
                     F656'  31  09A7  2192          BRW     CNX$SCAN_CSBS_FORK      ; Fork and resume scan
                                09AA  2193
                        05      09AA  2194 10$:     RSB
                                09AB  2195
                     F652'  31  09AB  2196 15$:     BRW     CNX$SCAN_CSBS_EXIT      ; Exit from iteration
                                09AE  2197
                     F64F'  31  09AE  2198 20$:     BRW     CNX$SCAN_CSBS_RETRY     ; Wait and resume scan
                                09B1  2199
                                09B1  2200 ;
                                09B1  2201 ; Describe joining system to all member systems
                                09B1  2202 ;
                     F64C'  30  09B1  2203 30$:     BSBW    CNX$SCAN_CSBS           ; Scan all CSBs
                   43 50    E9  09B4  2204          BLBC    R0,130$                 ; Branch if scan done
          1F 60 A3    11    E1  09B7  2205          BBC     #CSB$V_SELECTED, -      ; Branch if not selected
                                09BC  2206                  CSB$L_STATUS(R3),40$
          1A 60 A3    18    E0  09BC  2207          BBS     #CSB$V_LOCAL, -         ; Branch if local CSB
                                09C1  2208                  CSB$L_STATUS(R3),40$
          15 60 A3    00    E0  09C1  2209          BBS     #CSB$V_LONG_BREAK, -    ; Branch if connection permanently
                                09C6  2210                  CSB$L_STATUS(R3),40$    ;   broken
                     F637'  30  09C6  2211          BSBW    CNX$ALLOC_CDRP_ONLY     ; Get a fork block
                     F634'  30  09C9  2212          BSBW    CNX$RESOURCE_CHECK      ; Watch out for exhausted resources
                   0D 50    E9  09CC  2213          BLBC    R0,50$                  ; No memory available
                52   70 A4    D0  09CF  2214          MOVL    CLUB$L_CTX0(R4),R2      ; Address of joining node CSB
                   5A A4    B6  09D3  2215          INCW    CLUB$W_MSGCNT(R4)       ; Increment waiting count
                      07    10  09D6  2216          BSBB    60$                     ; Describe nodes
                     F625'  31  09D8  2217          BRW     CNX$SCAN_CSBS_FORK      ; Fork and resume scan
                                09DB  2218
                        05      09DB  2219 40$:     RSB
                                09DC  2220
                     F621'  31  09DC  2221 50$:     BRW     CNX$SCAN_CSBS_RETRY     ; Wait and resume scan
                                09DF  2222
                     0569   30  09DF  2223 60$:     BSBW    DESCRIBE_NODE           ; Tell one node about another
                54   64 A3    D0  09E2  2224          MOVL    CSB$L_CLUB(R3),R4       ; Address of CLUB
                   03 50    E9  09E6  2225          BLBC    R0,70$                  ; Error sending message
                   05 51    E8  09E9  2226          BLBS    R1,80$                  ; Branch if node is known
          00 1C A4    11    E2  09EC  2227 70$:     BBSS    #CLUB$V_UNLOCK, -       ; Request that all be unlocked
                                09F1  2228                  CLUB$L_FLAGS(R4),80$
                50      55  D0  09F1  2229 80$:     MOVL    R5,R0
              00000000'GF   16  09F4  2230          JSB     G^EXE$DEANONPAGED
                                09FA  2231 ;
```

```
                                09FA  2232 ; terminate threads -- except for the last one
                                09FA  2233 ;
                     5A A4   B7 09FA  2234 130$:   DECW    CLUB$W_MSGCNT(R4)
                        01   19 09FD  2235         BLSS    140$                    ; Branch when done
                             05 09FF  2236         RSB                             ; Terminate thread
                                0A00  2237
                      0831   30 0A00  2238 140$:   BSBW    CNX$CHECK_UNLOCK        ; Handle unlock request
                                0A03  2239 ;
                                0A03  2240 ; Now give the new node the information it needs to fill in its
                                0A03  2241 ; cluster vector.  The occupied slots will be taken care of in
                                0A03  2242 ; the normal course of events.  The available slots are given
                                0A03  2243 ; the value of the last sequence number used in this sequence
                                0A03  2244 ; of instructions.
                                0A03  2245 ;
              53  00000000'GF   3C 0A03  2246         MOVZWL  G^CLU$GW_MAXINDEX,R3    ; Number of vector slots
                        39   11 0A0A  2247         BRB     240$
                                0A0C  2248
                    F5F1'   30 0A0C  2249 200$:   BSBW    CNX$CLUB_WAIT           ; Wait a second, saving only R3
              51  00000000'GF   D0 0A0F  2250 210$:   MOVL    G^CLU$GL_CLUSVEC,R1     ; Address of cluster vector
                      6143   D5 0A16  2251         TSTL    (R1)[R3]                ; Contents of first slot
                        2A   19 0A19  2252         BLSS    240$                    ; Found CSB -- slot in use
                    F5E2'   30 0A1B  2253         BSBW    CNX$CLUB_FORK           ; Release control
                      0813   30 0A1E  2254         BSBW    CNX$CHECK_UNLOCK        ; Bail out if unlock has been requested
                    F5DC'   30 0A21  2255         BSBW    CNX$ALLOC_CDRP_ONLY     ; Get a fork block
                    F5D9'   30 0A24  2256         BSBW    CNX$RESOURCE_CHECK      ; Watch out for exhausted resources
                   E2 50    E9 0A27  2257         BLBC    R0,200$                 ; Branch if CDRP allocated
                      53    DD 0A2A  2258         PUSHL   R3                      ; Save scan context
                 53   70 A4  D0 0A2C  2259         MOVL    CLUB$L_CTX0(R4),R3      ; Address of joining node CSB
                    50   06  9A 0A30  2260         MOVZBL  #CLMCNX$K_FNC_VEC,R0    ; Cluster vector description
                      08DC   30 0A33  2261         BSBW    INIT_STD_MSG            ; Standard message initialization
           4C A5    0F04'CF   9E 0A36  2262         MOVAB   W^BLD_VEC_MSG, -        ; Vector message building routine
                                0A3C  2263                 CDRP$L_MSGBLD(R5)
                 48 A5   6E   D0 0A3C  2264         MOVL    (SP),CDRP$L_VAL8(R5)    ; Store Cluster vector index
                      F5BD'   30 0A40  2265         BSBW    CNX$SEND_FORGET         ; Queue message and forget it
                        08   BA 0A43  2266         POPR    #^M<R3>                 ; Restore context for scanning
                    C7 53    F4 0A45  2267 240$:   SOBGEQ  R3,210$                 ; Iterate over all slots
              54  00000000'GF   D0 0A48  2268         MOVL    G^CLU$GL_CLUB,R4        ; Address of CLUB
                                0A4F  2269 ;
                                0A4F  2270 ; Get here when all nodes have been described to all others
                                0A4F  2271 ; Now have told every node about every other node and each node's
                                0A4F  2272 ; CSB contains a map of nodes that can be seen
                                0A4F  2273 ;
                                0A4F  2274 JOIN_PROPOSE:
00EC C4   20   00   6E   00   2C 0A4F  2275         MOVC5   #0,(SP),#0, -           ; Init map of nodes that are
                                0A57  2276                 #CLUB$S_NODEMAP, -      ;   totally connected
                                0A57  2277                 CLUB$B_NODEMAP(R4)
                      F5A6'   30 0A57  2278         BSBW    CNX$SCAN_CSBS           ; Scan all CSB's
                    10 50    E9 0A5A  2279         BLBC    R0,240$                 ; Branch when done
                0A 60 A3    11   E1 0A5D  2280         BBC     #CSB$V_SELECTED, -      ; Branch if not selected
                                0A62  2281                 CSB$L_STATUS(R3),210$
                   50   4C A3   3C 0A62  2282         MOVZWL  CSB$W_CSID_IDX(R3),R0   ; CSID index of CSB
              00 00EC C4    50   E2 0A66  2283         BBSS    R0,CLUB$B_NODEMAP(R4), - ; Node is part of proposed cluster
                                0A6C  2284                 210$
                                05 0A6C  2285 210$:   RSB                             ; Return and resume scan
                                0A6D  2286
                                0A6D  2287 240$:
                      F590'   30 0A6D  2288         BSBW    CNX$DIRVEC_ADJ          ; Adjust size of lock manager directory vect
```

```
                     03 50    E8  0A70  2289          BLBS    R0,250$                 ; Branch on success
                      07D0    31  0A73  2290          BRW     UNLOCK_ALL              ; Can't adjust vector size, abort
                              0A76  2291 250$:
                              CA76  2292 ;
                              0A76  2293 ; Propose cluster -- Send Phase 1 Messages
                              0A76  2294 ;
                 70 A4  09    9A  0A76  2295          MOVZBL  #CLMCNX$K_FNC_JOIN, -   ; Facility specific message code
                              0A7A  2296                  CLUB$L_CTX0(R4)
              78 A4  0BFA'CF  9E  0A7A  2297          MOVAB   W^BLD_JOIN_MSG, -       ; Address of routine to build
                              0A80  2298                  CLUB$L_CTXT(R4)             ;  status message
                      05DC    30  0A80  2299          BSBW    SEND_PH1                ; Send phase 1 messages
                              0A83  2300 ;
                              0A83  2301 ; All nodes have seen Phase 1 and ACKed it.
                              0A83  2302 ; Now is the time to send Phase 2.
                              0A83  2303 ;
                      0638    31  0A83  2304          BRW     SEND_PH2
                              0A86  2305 ;
                              0A86  2306 ; At the completion of sending Phase 2 messages, this location
                              0A86  2307 ; is executed as a thread.
                              0A86  2308 ;
                              0A86  2309 JOIN_FINISH:
                              0A86  2310 FORM_JOIN_FINISH:
                   52  10 A4   D0  0A86  2311          MOVL    CLUB$L_LOCAL_CSB(R4),R2 ; Local node CSB
          008C C2  0CEC C4  20   28  0A8A  2312          MOVC3   #CSB$S_NODEMAP, -       ; Copy membership map from CLUB
                              0A92  2313                  CLUB$B_NODEMAP(R4), -       ;   to local node's CSB
                              0A92  2314                  CSB$B_NODEMAP(R2)
                      F56B'   30  0A92  2315          BSBW    CNX$SCAN_CSBS           ; Iterate over all CSBs
                    2A 50     E9  0A95  2316          BLBC    R0,20$                  ; Branch when done
                24 60 A3      11  0A98  2317          BBC     #CSB$V_SELECTED, -      ; Branch if not selected
                              0A9D  2318                  CSB$L_STATUS(R3),10$
                1F 60 A3      18  0A9D  2319          BBS     #CSB$V_LOCAL, -         ; Branch if local node
                              0AA2  2320                  CSB$L_STATUS(R3),10$
                         18   BB  0AA2  2321          PUSHR   #^M<R3,R4>              ; Save registers
          008C C3  00EC C4  20   28  0AA4  2322          MOVC3   #CSB$S_NODEMAP, -       ; Copy membership map from CLUB
                              0AAC  2323                  CLUB$B_NODEMAP(R4), -       ;   to selected node's CSB
                              0AAC  2324                  CSB$B_NODEMAP(R3)
                         18   BA  0AAC  2325          POPR    #^M<R3,R4>              ; Restore registers
                0E 60 A3      00  E1  0AAE  2326          BBC     #CSB$V_LONG_BREAK, -    ; Branch if no long break
                              0AB3  2327                  CSB$L_STATUS(R3),10$
                   52  10 A4   D0  0AB3  2328          MOVL    CLUB$L_LOCAL_CSB(R4),R2 ; Get local CSB address
                   50  4C A3   3C  0AB7  2329          MOVZWL  CSB$W_CSID_IDX(R3),R0   ; Remote node CSID index
              00 008C C2   50   E5  0ABB  2330          BBCC    R0,CSB$B_NODEMAP(R2),10$ ; Mark connection broken
                         05   RSB 0AC1  2331 10$:      RSB
                              0AC2  2332
                      F53B'   30  0AC2  2333 20$:      BSBW    CNX$FIX_EPID            ; Add node ID to EPIDs
                      089B    30  0AC5  2334          BSBW    UPDATE_QUORUM           ; Update quorum, votes, nodes
                      F535'   30  0AC8  2335          BSBW    CNX$SCAN_CSBS           ; Iterate over CSBs
                    28 50     E9  0ACB  2336          BLBC    R0,60$                  ; Branch when scan is done
                1D 60 A3      11  0ACE  2337          BBC     #CSB$V_SELECTED, -      ; Branch if CSB not selected
                              0AD3  2338                  CSB$L_STATUS(R3),40$
                13 60 A3      01  E0  0AD3  2339          BBS     #CSB$V_MEMBER, -        ; Skip if node is already a member
                              0AD8  2340                  CSB$L_STATUS(R3),30$
                      08D1    30  0AD8  2341          BSBW    ADD_NODE
                0B 60 A3      18  E1  0ADB  2342          BBC     #CSB$V_LOCAL, -         ; Branch if not local node
                              0AE0  2343                  CSB$L_STATUS(R3),30$        ;   and skip outputting message
                   55  53     D0  0AE0  2344          MOVL    R3,R5                   ; Address of CSB
                 50  0000'CF   9E  0AE3  2345          MOVAB   ADDNODE_MSG,R0          ; Node added to cluster message
```

```
                                                H 1

                     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
```

```
                                                                I  1
                     0B19  2370          .SBTTL  ADJUST_QUORUM - Adjust Cluster Quorum and/or Quorum Disk Membership
                     0B19  2371
                     0B19  2372 ;++
                     0B19  2373 ;
                     0B19  2374 ; FUNCTIONAL DESCRIPTION:
                     0B19  2375 ;
                     0B19  2376 ;       This routine adjusts the cluster quorum and/or changes quorum disk membershi
                     0B19  2377 ;       This is done in a synchronized manner using a 2 phase protocol.
                     0B19  2378 ;
                     0B19  2379 ; CALLING SEQUENCE:
                     0B19  2380 ;
                     0B19  2381 ;       JSB     ADJUST_QUORUM
                     0B19  2382 ;       IPL is IPL$_SCS
                     0B19  2383 ;
                     0B19  2384 ; INPUT PARAMETERS:
                     0B19  2385 ;
                     0B19  2386 ;       NONE
                     0B19  2387 ;
                     0B19  2388 ; OUTPUT PARAMETERS:
                     0B19  2389 ;
                     0B19  2390 ;       NONE
                     0B19  2391 ;
                     0B19  2392 ; COMPLETION CODES:
                     0B19  2393 ;
                     0B19  2394 ;       NONE
                     0B19  2395 ;
                     0B19  2396 ; SIDE EFFECTS:
                     0B19  2397 ;
                     0B19  2398 ;       R0-R5 are destroyed
                     0B19  2399 ;
                     0B19  2400 ;--
                     0B19  2401
                     0B19  2402 ADJUST_QUORUM:
           0085   30 0B19  2403          BSBW    INIT_TRANSITION         ; Initialization for becoming coordinator
      58 A4  04   90 0B1C  2404          MOVB    #CLM$CNX$K_XTN_QUORUM, - ; RECONFIGure transaction
                     0B20  2405                  CLUB$B_CUR_CODE(R4)
            55   D4 0B20  2406          CLRL    R5                      ; No CSB address
  50   0000'CF   9E 0B22  2407          MOVAB   QUORUM_MSG,R0           ; Initiating cluster reconfiguration message
          F4D6'  30 0B27  2408          BSBW    CNX$CONFIG_CHANGE       ; Make this state known
          F4D3'  30 0B2A  2409          BSBW    CNX$SCAN_CSBS
        0B 50    E9 0B2D  2410          BLBC    R0,20$                  ; Branch when done
05 60 A3   01    E1 0B30  2411          BBC     #CSB$V_MEMBER, -        ; Ignore nodes that are not members of
                     0B35  2412                  CSB$L_STATUS(R3),10$    ;   the local cluster
00 60 A3   11    E3 0B35  2413          BBCS    #CSB$V_SELECTED, -      ; Mark node selected
                     0B3A  2414                  CSB$L_STATUS(R3),10$
                05   0B3A  2415 10$:    RSB
                     0B3B  2416
                     0B3B  2417 20$:
                     0B3B  2418 ;
                     0B3B  2419 ; Selected nodes are all members of the cluster.
                     0B3B  2420 ;
          05F9   30 0B3B  2421          BSBW    LOCK_NODES              ; Attempt to lock selected nodes
                     0B3E  2422 ;
                     0B3E  2423 ; Get here when all nodes in the cluster have been locked.
                     0B3E  2424 ; Compute and send Phase 1 proposal messages.
                     0B3E  2425 ;
          F4BF'  30 0B3E  2426          BSBW    CNX$QUORUM_CALC         ; Calculate and store quorum related paramet
```

```
                     0B41  2427
                                                                ; R0 is computed number of votes
      25 1C A4   1B  E1  0B41  2428          BBC     #CLUB$V_ADJ_QUORUM, -    ; Branch if no quorum adjustment requested
                     0B46  2429                  CLUB$L_FLAGS(R4),60$
00A4 C4    00A6 C4  B0  0B46  2430          MOVW    CLUB$W_ADJ_QUORUM(R4), - ; Copy proposed value to new cell
                     0B4D  2431                  CLUB$W_NEWQUORUM(R4)
      50   00A6 C4  B1  0B4D  2432          CMPW    CLUB$W_ADJ_QUORUM(R4),R0 ; Minimize value with number of votes
                05   1F  0B52  2433          BLSSU   30$                      ; Branch if new value would
      00A4 C4    50  B0  0B54  2434          MOVW    R0,CLUB$W_NEWQUORUM(R4) ; Use current number of votes
           50    02  C0  0B59  2435 30$:     ADDL2   #2,R0                    ; Compute (votes+2)/2
           50    02  C6  0B5C  2436          DIVL2   #2,R0
      50   00A4 C4  B1  0B5F  2437          CMPW    CLUB$W_NEWQUORUM(R4),R0 ; New quorum >= (votes+2)/2?
                05   1E  0B64  2438          BGEQU   40$                      ; Branch if yes
      00A4 C4    50  B0  0B66  2439          MOVW    R0,CLUB$W_NEWQUORUM(R4)
                     0B68  2440 40$:
                     0B68  2441 60$:
                     0B68  2442 ;
                     0B68  2443 ; Propose cluster -- Send Phase 1 Messages
                     0B68  2444 ;
        70 A4    0D  9A  0B68  2445          MOVZBL  #CLMCNX$K_FNC_QUORUM, - ; Message code
                     0B6C  2446                  CLUB$L_CTX0(R4)
   78 A4    0BFA'CF  9E  0B6C  2447          MOVAB   W^BLD_QUORUM_MSG, -      ; Address of routine to build
                     0B72  2448                  CLUB$L_CTX1(R4)              ;  status message
            04E7     30  0B72  2449          BSBW    SEND_PH1                 ; Send phase 1 messages
                     0B75  2450 ;
                     0B75  2451 ; Phase 1 has been sent.  Nothing can stop us now short of our own
                     0B75  2452 ; failure.  Therefore, it is time to clear the bit requesting the
                     0B75  2453 ; quorum adjustment.  This node has done its best...
                     0B75  2454 ;
      00 1C A4   1B  E5  0B75  2455          BBCC    #CLUB$V_ADJ_QUORUM, -    ; Clear quorum adjustment flag
                     0B7A  2456                  CLUB$L_FLAGS(R4),100$
                     0B7A  2457 100$:
                     0B7A  2458 ;
                     0B7A  2459 ; All nodes have seen Phase 1 and ACKed it.
                     0B7A  2460 ; Now is the time to send Phase 2.
                     0B7A  2461 ;
            053E     31  0B7A  2462          BRW     SEND_PH2
                     0B7D  2463 ;
                     0B7D  2464 ; At the completion of sending Phase 2 messages, this location
                     0B7D  2465 ; is executed as a thread.
                     0B7D  2466 ;
                     0B7D  2467 QUORUM_FINISH:
            07E0     30  0B7D  2468          BSBW    UPDATE_QUORUM            ; Update quorum, votes, nodes
                     0B80  2469 ;
                     0B80  2470 ; Reduce quorum in CSB's to prevent cluster quorum from ratcheting right
                     0B80  2471 ; back up with the next state transition
                     0B80  2472 ;
            F47A'    30  0B80  2473          BSBW    CNX$SCAN_CSBS            ; Iterate over CSBs
          15 50      E9  0B83  2474          BLBC    R0,230$                  ; Branch when scan is done
0F 60 A3    01      E1  0B86  2475          BBC     #CSB$V_MEMBER, -         ; Branch if node is cluster member
                     0B8B  2476                  CSB$L_STATUS(R3), 220$
20 A4    52 A3      B1  0B8B  2477          CMPW    CSB$W_QUORUM(R3), -      ; Compare sysgen parameter with current
                     0B90  2478                  CLUB$W_QUORUM(R4)        ;  cluster quorum
             05      1B  0B90  2479          BLEQU   210$                     ; Branch if less than cluster quorum
52 A3    20 A4      B0  0B92  2480          MOVW    CLUB$W_QUORUM(R4), -     ; Lower copy of sysgen parameter to match
                     0B97  2481                  CSB$W_QUORUM(R3)         ;  new, reduced cluster quorum
            F463'    30  0B97  2482 210$:    BSBW    CNX$MARK_UNLOCKED        ; Mark node no longer locked
                     05  0B9A  2483 220$:    RSB                              ; Continue iteration
```

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

```
                     0BA1  2487          .SBTTL  INIT_TRANSITION - Initialization for a transition
                     0BA1  2488
                     0BA1  2489 ;++
                     0BA1  2490 ;
                     0BA1  2491 ; FUNCTIONAL DESCRIPTION:
                     0BA1  2492 ;
                     0BA1  2493 ;       This routine is the common initialization code for all
                     0BA1  2494 ;       transitions.  It is executed by any node which is attempting
                     0BA1  2495 ;       to become the coordinator.
                     0BA1  2496 ;
                     0BA1  2497 ; CALLING SEQUENCE:
                     0BA1  2498 ;
                     0BA1  2499 ;       JSB     INIT_TRANSITION
                     0BA1  2500 ;       IPL is IPL$_SCS
                     0BA1  2501 ;
                     0BA1  2502 ; INPUT PARAMETERS:
                     0BA1  2503 ;
                     0BA1  2504 ;       NONE
                     0BA1  2505 ;
                     0BA1  2506 ; OUTPUT PARAMETERS:
                     0BA1  2507 ;
                     0BA1  2508 ;       R3:     Address of local CSB
                     0BA1  2509 ;       R4:     Address of CLUB
                     0BA1  2510 ;
                     0BA1  2511 ; COMPLETION CODES:
                     0BA1  2512 ;
                     0BA1  2513 ;       NONE
                     0BA1  2514 ;
                     0BA1  2515 ; SIDE EFFECTS:
                     0BA1  2516 ;
                     0BA1  2517 ;       R0-R2,R5 are destroyed
                     0BA1  2518 ;
                     0BA1  2519 ;--
                     0BA1  2520
                     0BA1  2521 INIT_TRANSITION:
   54   000000 '''GF   D0  0BA1  2522          MOVL    G^CLU$GL_CLUB,R4        ; Address of CLUB
1C A4   20000000 8F   C8  0BA8  2523          BISL2   #CLUB$M_TRANSITION, -   ; Mark transition started
                     0BB0  2524                  CLUB$L_FLAGS(R4)
         0084 C4     D4  0BB0  2525          CLRL    CLUB$L_TQE(R4)          ; Abandon any timeout in progress
      50   00CC C4     9E  0BB4  2526          MOVAB   CLUB$B_FORK_BLOCK(R4),R0 ; Address of transition fork block
   03 1C A0    00     E5  0BB9  2527          BBCC    #CLUBFKB$V_FKB_BUSY, -  ; Branch if fork block is not
                     0BBE  2528                  CLUBFKB$L_STATUS(R0),10$ ;  busy and mark no busy
         50    60     0F  0BBE  2529          REMQUE  CLUBFKB$B_FORK_BLOCK(R0),R0 ; Remove from queue
1C A4   00020000 8F   CA  0BC1  2530 10$:     BICL2   #CLUB$M_UNLOCK, -       ; Clear unlock request
                     0BC9  2531                  CLUB$L_FLAGS(R4)
            68 A4    D6  0BC9  2532 20$:     INCL    CLUB$L_MAX_XTN(R4)      ; Compute transaction ID
               FB    13  0BCC  2533          BEQL    20$                     ; Disallow transition 0
   48 A4    68 A4    D0  0BCE  2534          MOVL    CLUB$L_MAX_XTN(R4), -
                     0BD3  2535                  CLUB$L_CUR_XTN(R4)
      53    10 A4    D0  0BD3  2536          MOVL    CLUB$L_LOCAL_CSB(R4),R3 ; CSB for this node
   4C A4    4C A3    D0  0BD7  2537          MOVL    CSB$L_CSID(R3), -       ; CSID of coordinator
                     0BDC  2538                  CLUB$L_CUR_COORD(R4)
      5C A4    53    D0  0BDC  2539          MOVL    R3,CLUB$L_COORD(R4)     ; Mark this node the coordinator
      59 A4    20    90  0BE0  2540          MOVB    #CLMCNX$K_LOCK, -       ; Set transaction phase
                     0BE4  2541                  CLUB$B_CUR_PHASE(R4)
50 A4   00000000'GF   7D  0BE4  2542          MOVQ    G^EXE$GQ_SYSTIME, -     ; Current time is timestamp for
                     0BEC  2543                  CLUB$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  2551          .SBTTL  BLD_FORM_MSG - Build Message Proposing a New Cluster
                        0BFA  2552          .SBTTL  BLD_RECONFIG_MSG - Build Message Proposing a Reconfigured Cluster
                        0BFA  2553          .SBTTL  BLD_JOIN_MSG - Build Message Proposing a New Node in a Cluster
                        0BFA  2554          .SBTTL  BLD_QUORUM_MSG - Build Message Proposing a Quorum/Quorum Disk Chanag
                        0BFA  2555 ;++
                        0BFA  2556 ;
                        0BFA  2557 ; FUNCTIONAL DESCRIPTION:
                        0BFA  2558 ;
                        0BFA  2559 ;       This routine builds a message proposing the formation, reconfiguration, or
                        0BFA  2560 ;       addition of a node to a cluster.
                        0BFA  2561 ;       It uses the standard message building routine to build the message header.
                        0BFA  2562 ;
                        0BFA  2563 ; CALLING SEQUENCE:
                        0BFA  2564 ;
                        0BFA  2565 ;       JSB     BLD_FORM_MSG
                        0BFA  2566 ;       JSB     BLD_RECONFIG_MSG
                        0BFA  2567 ;       JSB     BLD_JOIN_MSG
                        0BFA  2568 ;       JSB     BLD_QUORUM_MSG
                        0BFA  2569 ;       IPL is IPL$_SCS
                        0BFA  2570 ;
                        0BFA  2571 ; INPUT PARAMETERS:
                        0BFA  2572 ;
                        0BFA  2573 ;       R2:     Address of message buffer
                        0BFA  2574 ;       R3:     Address of CSB
                        0BFA  2575 ;       R4:     Address of PDT
                        0BFA  2576 ;       R5:     Address of CDRP
                        0BFA  2577 ;               CDRP$L_VAL1(R5): Byte 0 contains facility code (negated for response
                        0BFA  2578 ;                                Byte 1 contains facility specific function code
                        0BFA  2579 ;               CDRP$L_VAL2(R5): Transition ID (from CLUB$L_CUR_XTN)
                        0BFA  2580 ;               CDRP$L_VAL3(R5): Byte 0 contains transition phase (from CLUB$B_CUR_P
                        0BFA  2581 ;                                Byte 1 contains transition code (from CLUB$B_XTN_CO
                        0BFA  2582 ;                                Byte 2 contains success/failure flag
                        0BFA  2583 ;                                Byte 3 contains reply code
                        0BFA  2584 ;
                        0BFA  2585 ; OUTPUT PARAMETERS:
                        0BFA  2586 ;
                        0BFA  2587 ;       NONE
                        0BFA  2588 ;
                        0BFA  2589 ; COMPLETION CODES:
                        0BFA  2590 ;
                        0BFA  2591 ;       NONE
                        0BFA  2592 ;
                        0BFA  2593 ; SIDE EFFECTS:
                        0BFA  2594 ;
                        0BFA  2595 ;       R0 and R1 are destroyed
                        0BFA  2596 ;
                        0BFA  2597 ;--
                        0BFA  2598
                        0BFA  2599 BLD_FORM_MSG:
                        0BFA  2600 BLD_RECONFIG_MSG:
                        0BFA  2601 BLD_JOIN_MSG:
                        0BFA  2602 BLD_QUORUM_MSG:
              073C   30 0BFA  2603          BSBW    BLD_STD_MSG             ; Fill in standard fields
          007C 8F    BB 0BFD  2604          PUSHR   #^M<R2,R3,R4,R5,R6>     ; Save registers
       56   64 A3    D0 0C01  2605          MOVL    CSB$L_CLUB(R3),R6       ; Address of CLUB
 14 A2   64 A6    B0 0C05  2606          MOVW    CLUB$Q_NEXT_CSID(R6), - ; Save CSID assignment context
                     0C0A  2607                  CLMPRO$W_NEXT_CSID(R2)
```

```
                                                B 2
                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  2635          .SBTTL  CNX$RCVD_FORM - Cluster formation proposal received
                        0C4F  2636
                        0C4F  2637 ;++
                        0C4F  2638 ;
                        0C4F  2639 ; FUNCTIONAL DESCRIPTION:
                        0C4F  2640 ;
                        0C4F  2641 ;       This routine is called cluster formation proposal message is received.
                        0C4F  2642 ;       A response must be sent which ACKs or NAKs the request.  The
                        0C4F  2643 ;       request is NAKd if this node does not have a connection to all proposed
                        0C4F  2644 ;       member nodes.
                        0C4F  2645 ;
                        0C4F  2646 ; CALLING SEQUENCE:
                        0C4F  2647 ;
                        0C4F  2648 ;       JSB     CNX$RCVD_FORM
                        0C4F  2649 ;       IPL is IPL$_SCS
                        0C4F  2650 ;
                        0C4F  2651 ; INPUT PARAMETERS:
                        0C4F  2652 ;
                        0C4F  2653 ;       R2:     Message address
                        0C4F  2654 ;       R3:     CSB of sending system
                        0C4F  2655 ;       R4:     PDT address
                        0C4F  2656 ;       R5:     CDRP address (uninitialized)
                        0C4F  2657 ;
                        0C4F  2658 ; OUTPUT PARAMETERS:
                        0C4F  2659 ;
                        0C4F  2660 ;       NONE
                        0C4F  2661 ;
                        0C4F  2662 ; COMPLETION CODES:
                        0C4F  2663 ;
                        0C4F  2664 ;       NONE
                        0C4F  2665 ;
                        0C4F  2666 ; SIDE EFFECTS:
                        0C4F  2667 ;
                        0C4F  2668 ;       R0-R5 may be destroyed.
                        0C4F  2669 ;--
                        0C4F  2670
                        0C4F  2671 CNX$RCVD_FORM::
                  2C    BB    0C4F  2672          PUSHR   #^M<R2,R3,R5>           ; Save needed registers
              F3AC'     30    0C51  2673          BSBW    CNX$INIT_CDRP           ; Initialize the CDRP for the response
           52   6E     D0    0C54  2674          MOVL    (SP),R2                 ; Restore message address
              06EF     30    0C57  2675          BSBW    MSG_CHECK               ; Validate message
     64 A4    14 A2   B0    0C5A  2676          MOVW    CLMPRO$W_NEXT_CSID(R2), - ; Save CSID assignment context
                        0C5F  2677                  CLUB$W_NEXT_CSID(R4)
   00A4 C4    16 A2   B0    0C5F  2678          MOVW    CLMPRO$W_QUORUM(R2), -  ; Cluster quorum
                        0C65  2679                  CLUB$W_NEWQUORUM(R4)
   00AC C4    18 A2   B0    0C65  2680          MOVW    CLMPRO$W_MEMSEQ(R2), -  ; Cluster membership state sequence
                        0C6B  2681                  CLUB$W_MEMSEQ(R4)       ;   number
   00A8 C4    1C A2   D0    0C6B  2682          MOVL    CLMPRO$L_FMERIT(R2), -  ; Proposed cluster figure of merit
                        0C71  2683                  CLUB$L_FMERIT(R4)
     2C A4    20 A2   7D    0C71  2684          MOVQ    CLMPRO$Q_FTIME(R2), -   ; Cluster foundation time
                        0C76  2685                  CLUB$Q_FTIME(R4)
   0094 C4    28 A2   7D    0C76  2686          MOVQ    CLMPRO$Q_CURTIME(R2), - ; Save time from message
                        0C7C  2687                  CLUB$Q_NEWTIME(R4)
     46 A4    1A A2   B0    0C7C  2688          MOVW    CLMPRO$W_QDVOTES(R2), - ; Quorum disk votes
                        0C81  2689                  CLUB$W_NEWQDVOTES(R4)
 1C A4    04000000 8F   CA    0C81  2690          BICL    #CLUB$M_QF_NEWVOTE, -   ; Assume no quorum disk membership
                        0C89  2691                  CLUB$L_FLAGS(R4)
```

```
                0C 30 A2    00    E1   0C89  2692          BBC     #CLMPRO$V_QF_VOTE, -    ; Branch if no quorum disk membership
                                      0C8E  2693                  CLMPRO$B_FLAGS(R2),10$
                            51    D4   0C8E  2694          CLRL    R1                      ; Assume failure
                4A 1C A4    01    E1   0C90  2695          BBC     #CLUB$V_QF_ACTIVE, -    ; Branch if disk not locally active
                                      0C95  2696                  CLUB$L_FLAGS(R4),50$    ;   and reject proposal
                00 1C A4    1A    E2   0C95  2697          BBSS    #CLUB$V_QF_NEWVOTE, -   ; Set quorum disk membership
                                      0C9A  2698                  CLUB$L_FLAGS(R4),10$
                            1C    BB   0C9A  2699 10$:      PUSHR   #^M<R2,R3,R4>           ; Save registers
            26 A4   31 A2   06    28   0C9C  2700          MOVC3   #CLUB$S_FSYSID, -       ; Cluster founding system ID
                                      0CA2  2701                  CLMPRO$B_FSYSID(R2), -
                                      0CA2  2702                  CLUB$B_FSYSID(R4)
                            1C    BA   0CA2  2703          POPR    #^M<R2,R3,R4>           ; Restore registers
                            1C    BB   0CA4  2704          PUSHR   #^M<R2,R3,R4>           ; Save registers
         00EC C4    37 A2   20    28   0CA6  2705          MOVC3   #CLUB$S_NODEMAP, -      ; Save map of cluster nodes
                                      0CAD  2706                  CLMPRO$B_NODEMAP(R2), - ;   in CLUB
                                      0CAD  2707                  CLUB$B_NODEMAP(R4)
                            1C    BA   0CAD  2708          POPR    #^M<R2,R3,R4>           ; Restore registers
         009C C4    00000000'GF   7D   0CAF  2709          MOVQ    G^EXE$GQ_SYSTIME, -     ; Save current system time
                                      0CB8  2710                  CLUB$Q_NEWTIME_REF(R4)  ;   as a reference
                        51    01    D0   0CB8  2711          MOVL    #1,R1                   ; Anticipate success
                           F342'    30   0CBB  2712          BSBW    CNX$SCAN_CSBS           ; Iterate over all CSBs
                         1E 50      E9   0CBE  2713          BLBC    R0,50$                  ; Branch when done
                18 60 A3    11    E1   0CC1  2714          BBC     #CSB$V_SELECTED, -      ; Branch if node not selected
                                      0CC6  2715                  CSB$L_STATUS(R3),40$
                    50    4C A3    3C   0CC6  2716          MOVZWL  CSB$W_CSID_IDX(R3),R0   ; CSID index
                0A 37 A2    50    E5   0CCA  2717          BBCC    R0, -                   ; Branch if node is not in proposed
                                      0CCF  2718                  CLMPRO$B_NODEMAP(R2), - ;   cluster
                                      0CCF  2719                  30$
                0A 60 A3    00    E1   0CCF  2720          BBC     #CSB$V_LONG_BREAK, -    ; Branch if no long break seen
                                      0CD4  2721                  CSB$L_STATUS(R3),40$
                            51    D4   0CD4  2722          CLRL    R1
                       F327'      31   0CD6  2723          BRW     CNX$SCAN_CSBS_EXIT      ; Exit from CSB scanning loop
                                      0CD9  2724
             00 60 A3    11    E4   0CD9  2725 30$:       BBSC    #CSB$V_SELECTED, -      ; Clear selected bit
                                      0CDE  2726                  CSB$L_STATUS(R3),40$
                            05    0CDE  2727 40$:       RSB
                                      0CDF  2728
                            51    DD   0CDF  2729 50$:      PUSHL   R1                      ; Save status
        37 A2       20      00    3B   0CE1  2730          SKPC    #0,#CLMPRO$S_NODEMAP, - ; Look for any missing nodes
                                      0CE6  2731                  CLMPRO$B_NODEMAP(R2)
                            07    13   0CE6  2732          BEQL    60$                     ; All nodes accounted for
             02 1C A4       11    E1   0CE8  2733          BBC     #CLUB$V_UNLOCK, -       ; Branch if no unlock request
                                      0CED  2734                  CLUB$L_FLAGS(R4), 60$   ;   pending
                            6E    D4   0CED  2735          CLRL    (SP)                    ; NAK request
                      09 6E     E9   0CEF  2736 60$:      BLBC    (SP),70$                ; Branch on NAK pending
                       F30B'    30   0CF2  2737          BSBW    CNX$DIRVEC_ADJ          ; Adjust size of lock manager directory vect
                    03 50       E8   0CF5  2738          BLBS    R0,70$                  ; Branch on success
                   6E   50      D0   0CF8  2739          MOVL    R0,(SP)                 ; Make sure proposal is NAKed
                        2D      BA   0CFB  2740 70$:      POPR    #^M<R0,R2,R3,R5>        ; Restore needed registers
                        50      DD   0CFD  2741          PUSHL   R0                      ; Save status
                   50   07      9A   0CFF  2742          MOVZBL  #CLMCNX$K_FNC_FORM,R0   ; Facility specific function code
                      0604      30   0D02  2743          BSBW    CNX$INIT_STD_RESP       ; Init CDRP for standard response
                36 A5   8E      F6   0D05  2744          CVTLB   (SP)+,CDRP$L_VAL3+2(R5) ; Store success/failure flag
                       F2F4'    31   0D09  2745          BRW     CNX$RESP_FORGET         ; Queue response message
```

```
                        0D0C  2747          .SBTTL  CNX$RCVD_RECONFIG - Cluster Reconfiguration proposal received
                        0D0C  2748
                        0D0C  2749 ;++
                        0D0C  2750 ;
                        0D0C  2751 ; FUNCTIONAL DESCRIPTION:
                        0D0C  2752 ;
                        0D0C  2753 ;       This routine is called when a cluster reconfiguration proposal
                        0D0C  2754 ;       message is received.
                        0D0C  2755 ;       A response must be sent which ACKs or NAKs the request.  The
                        0D0C  2756 ;       request is NAKd if this node does not have a connection to exactly
                        0D0C  2757 ;       the set of member nodes of the proposed reconfigured cluster.
                        0D0C  2758 ;
                        0D0C  2759 ; CALLING SEQUENCE:
                        0D0C  2760 ;
                        0D0C  2761 ;       JSB     CNX$RCVD_RECONFIG
                        0D0C  2762 ;       IPL is IPL$_SCS
                        0D0C  2763 ;
                        0D0C  2764 ; INPUT PARAMETERS:
                        0D0C  2765 ;
                        0D0C  2766 ;       R2:     Message address
                        0D0C  2767 ;       R3:     CSB of sending system
                        0D0C  2768 ;       R4:     PDT address
                        0D0C  2769 ;       R5:     CDRP address (uninitialized)
                        0D0C  2770 ;
                        0D0C  2771 ; OUTPUT PARAMETERS:
                        0D0C  2772 ;
                        0D0C  2773 ;       NONE
                        0D0C  2774 ;
                        0D0C  2775 ; COMPLETION CODES:
                        0D0C  2776 ;
                        0D0C  2777 ;       NONE
                        0D0C  2778 ;
                        0D0C  2779 ; SIDE EFFECTS:
                        0D0C  2780 ;
                        0D0C  2781 ;       R0-R5 may be destroyed.
                        0D0C  2782 ;--
                        0D0C  2783
                        0D0C  2784 CNX$RCVD_RECONFIG::
                  2C    BB    0D0C  2785          PUSHR   #^M<R2,R3,R5>           ; Save needed registers
              F2EF'     30    0D0E  2786          BSBW    CNX$INIT_CDRP           ; Initialize the CDRP for the response
           52   6E     7D    0D11  2787          MOVQ    (SP),R2                 ; Restore message address
              0632     30    0D14  2788          BSBW    MSG_CHECK               ; Validate message
 1C A4    00800000    8F    C8    0D17  2789          BISL2   #CLUB$M_LOST_CNX, -     ; Mark connection to member lost
                              0D1F  2790                  CLUB$L_FLAGS(R4)
    00000000'GF   01    8E    0D1F  2791          MNEGB   #1,G^LCK$GB_STALLREQS   ; Stall lock requests
     00A8 C4   1C A2    D0    0D26  2792          MOVL    CLMPRO$L_FMERIT(R2), -  ; Proposed cluster figure of merit
                              0D2C  2793                  CLUB$L_FMERIT(R4)
00EC C4   37 A2    20    28    0D2C  2794          MOVC3   #CLMPRO$S_NODEMAP, -    ; Save map of cluster nodes
                              0D33  2795                  CLMPRO$B_NODEMAP(R2), - ;   in CLUB
                              0D33  2796                  CLUB$B_NODEMAP(R4)
                              0D33  2797
           51   01    D0    0D33  2798          MOVL    #1,R1                   ; Status indicator
           52   6E    D0    0D36  2799          MOVL    (SP),R2                 ; Message buffer address
              F2C4'     30    0D39  2800          BSBW    CNX$SCAN_CSBS           ; Iterate over all CSBs
             1E 50      E9    0D3C  2801          BLBC    R0,20$                  ; Branch when iteration is complete
       18 60 A3   01    E1    0D3F  2802          BBC     #CSB$V_MEMBER, -        ; Branch if node is not a cluster member
                              0D44  2803                  CSB$L_STATUS(R3),10$
```

```
                60 A3   00020000 8F    C8  0D44  2804          BISL2   #CSB$M_SELECTED, -      ; Select every member
                                      0D4C  2805                        CSB$L_STATUS(R3)
                     50   4C A3    3C  0D4C  2806          MOVZWL  CSB$W_CSID_IDX(R3),R0   ; System index
                07 37 A2    50    E5  0D50  2807          BBCC    R0, -                   ; Branch if member not in proposed cluster
                                      0D55  2808                        CLMPRO$B_NODEMAP(R2),10$
                02 60 A3    00    E1  0D55  2809          BBC     #CSB$V_LONG_BREAK, -    ; Branch if no long break seen
                                      0D5A  2810                        CSB$L_STATUS(R3),10$
                            51    D4  0D5A  2811          CLRL    R1                      ; Force failure of proposal
                            05    0D5C  2812 10$:         RSB
                                      0D5D  2813
                         26 51    E9  0D5D  2814 20$:     BLBC    R1,25$                  ; Branch on previous decision to reject
                00A4 C4    16 A2   B0  0D60  2815          MOVW    CLMPRO$W_QUORUM(R2), -  ; Store quorum
                                      0D66  2816                        CLUB$W_NEWQUORUM(R4)
                  46 A4    1A A2   B0  0D66  2817          MOVW    CLMPRO$W_QDVOTES(R2), - ; Store quorum disk votes
                                      0D6B  2818                        CLUB$W_NEWQDVOTES(R4)
            1C A4   04000000 8F    CA  0D6B  2819          BICL    #CLUB$M_QF_NEWVOTE, -   ; Assume no quorum disk membership
                                      0D73  2820                        CLUB$L_FLAGS(R4)
                  0A 30 A2    00    E1  0D73  2821          BBC     #CLMPRO$V_QF_VOTE, -    ; Branch if no quorum disk membership
                                      0D78  2822                        CLMPRO$B_FLAGS(R2),23$
                  07 1C A4    01    E1  0D78  2823          BBC     #CLUB$V_QF_ACTIVE, -    ; Branch if disk not locally active
                                      0D7D  2824                        CLUB$L_FLAGS(R4),24$    ;   and reject proposal
                  00 1C A4    1A    E2  0D7D  2825          BBSS    #CLUB$V_QF_NEWVOTE, -   ; Set quorum disk membership
                                      0D82  2826                        CLUB$L_FLAGS(R4),23$
                            02    11  0D82  2827 23$:     BRB     25$
                                      0D84  2828
                            51    D4  0D84  2829 24$:     CLRL    R1                      ; Reject proposal
                            51    DD  0D86  2830 25$:     PUSHL   R1                      ; Save status return
                37 A2    20   00    3B  0D88  2831          SKPC    #0,#CLMPRO$S_NODEMAP, - ; Look for any missing nodes
                                      0D8D  2832                        CLMPRO$B_NODEMAP(R2)
                            04    13  0D8D  2833          BEQL    30$                     ; All nodes accounted for
                                      0D8F  2834          BUG_CHECK       CNXMGRERR,FATAL ; Some non-member node proposed as member
                                      0D93  2835
                      31 6E    E9  0D93  2836 30$:     BLBC    (SP),70$                ; Reject proposal as inconsistent with local
              2A 1C A4    11    E0  0D96  2837          BBS     #CLUB$V_UNLOCK, -       ; Branch if unlock request
                                      0D9B  2838                        CLUB$L_FLAGS(R4),60$    ;   is pending
                    F262'   30    0D9B  2839          BSBW    CNX$OPT                 ; Try to compute better subcluster
                   03 50    E9  0D9E  2840          BLBC    R0,40$                  ; Branch on computation failure
                   21 51    E8  0DA1  2841          BLBS    R1,60$                  ; Branch if a better cluster was computed
                    F259'   30    0DA4  2842 40$:     BSBW    CNX$DIRVEC_ADJ          ; Adjust size of lock manager directory vect
                   1B 50    E9  0DA7  2843          BLBC    R0,60$                  ; Branch on failure
                    F253'   30    0DAA  2844          BSBW    CNX$SCAN_CSBS           ; Iterate over all CSBs
                   17 50    E9  0DAD  2845          BLBC    R0,70$                  ; Branch when iteration done
              0F 60 A3    01    E1  0DB0  2846          BBC     #CSB$V_MEMBER, -        ; Branch if system is not a member
                                      0DB5  2847                        CSB$L_STATUS(R3),50$
                50   4C A3    3C  0DB5  2848          MOVZWL  CSB$W_CSID_IDX(R3),R0   ; System ID index
          05 00EC C4    50    E0  0DB9  2849          BBS     R0,CLUB$B_NODEMAP(R4),50$ ; Branch if system is in proposed cluster
              00 60 A3    11    E5  0DBF  2850          BBCC    #CSB$V_SELECTED, -      ; Clear selected bit
                                      0DC4  2851                        CSB$L_STATUS(R3),50$
                            05    0DC4  2852 50$:     RSB                             ; End of iteration step
                                      0DC5  2853
                            6E    D4  0DC5  2854 60$:     CLRL    (SP)                    ; Failure -- reject proposal
                            2D    BA  0DC7  2855 70$:     POPR    #^M<R0,R2,R3,R5>        ; Restore status and needed registers
                            50    DD  0DC9  2856          PUSHL   R0                      ; Save status
                      50    08    9A  0DCB  2857          MOVZBL  #CLMCNX$K_FNC_RECONFIG,R0       ; Facility specific function code
                         0538   30    0DCE  2858          BSBW    CNX$INIT_STD_RESP       ; Init CDRP for standard response
                36 A5    8E    F6  0DD1  2859          CVTLB   (SP)+,CDRP$L_VAL3+2(R5) ; Store success/failure flag
                    F228'   31    0DD5  2860          BRW     CNX$RESP_FORGET         ; Queue response message
```

```
                                0DD8  2862          .SBTTL  CNX$RCVD_JOIN - Cluster node addition proposal received
                                0DD8  2863
                                0DD8  2864 ;++
                                0DD8  2865 ;
                                0DD8  2866 ; FUNCTIONAL DESCRIPTION:
                                0DD8  2867 ;
                                0DD8  2868 ;       This routine is called cluster node aaddition proposal message is received.
                                0DD8  2869 ;       A response must be sent which ACKs or NAKs the request.  The
                                0DD8  2870 ;       request is NAKd if this node does not have a connection to all proposed
                                0DD8  2871 ;       member nodes.
                                0DD8  2872 ;
                                0DD8  2873 ; CALLING SEQUENCE:
                                0DD8  2874 ;
                                0DD8  2875 ;       JSB     CNX$RCVD_JOIN
                                0DD8  2876 ;       IPL is IPL$_SCS
                                0DD8  2877 ;
                                0DD8  2878 ; INPUT PARAMETERS:
                                0DD8  2879 ;
                                0DD8  2880 ;       R2:     Message address
                                0DD8  2881 ;       R3:     CSB of sending system
                                0DD8  2882 ;       R4:     PDT address
                                0DD8  2883 ;       R5:     CDRP address (uninitialized)
                                0DD8  2884 ;
                                0DD8  2885 ; OUTPUT PARAMETERS:
                                0DD8  2886 ;
                                0DD8  2887 ;       NONE
                                0DD8  2888 ;
                                0DD8  2889 ; COMPLETION CODES:
                                0DD8  2890 ;
                                0DD8  2891 ;       NONE
                                0DD8  2892 ;
                                0DD8  2893 ; SIDE EFFECTS:
                                0DD8  2894 ;
                                0DD8  2895 ;       R0-R5 may be destroyed.
                                0DD8  2896 ;--
                                0DD8  2897
                                0DD8  2898 CNX$RCVD_JOIN::
                          00    DD  0DD8  2899          PUSHL   #0                      ; Anticipate NAK
                          2C    BB  0DDA  2900          PUSHR   #^M<R2,R3,R5>           ; Save needed registers
                      F221'     30  0DDC  2901          BSBW    CNX$INIT_CDRP           ; Initialize the CDRP for the response
                   52   6E     D0  0DDF  2902          MOVL    (SP),R2                 ; Restore message address
                      0564     30  0DE2  2903          BSBW    MSG_CHECK               ; Validate message
          03 1C A4     17     E1  0DE5  2904          BBC     #CLUB$V_LOST_CNX, -     ; Branch if node addition not inhibited
                                0DEA  2905                  CLUB$L_FLAGS(R4),10$
                        00A0    31  0DEA  2906          BRW     70$                     ; Helper branch when node addition inhibited
          64 A4      14 A2     B0  0DED  2907 10$:     MOVW    CLMPRO$W_NEXT_CSID(R2), - ; Save CSID assignment context
                                0DF2  2908                  CLUB$W_NEXT_CSID(R4)
       00AC C4       18 A2     B0  0DF2  2909          MOVW    CLMPRO$W_MEMSEQ(R2), -  ; Cluster membership state sequence
                                0DF8  2910                  CLUB$W_MEMSEQ(R4)       ;   number
          1A 1C A4     00     E0  0DF8  2911          BBS     #CLUB$V_CLUSTER, -      ; Branch if node is already a cluster member
                                0DFD  2912                  CLUB$L_FLAGS(R4),30$
                                0DFD  2913 ;
                                0DFD  2914 ; Receiving node is not cluster member
                                0DFD  2915 ;
          2C A4      20 A2     7D  0DFD  2916          MOVQ    CLMPRO$Q_FTIME(R2), -   ; Cluster foundation time
                                0E02  2917                  CLUB$Q_FTIME(R4)
       0094 C4       28 A2     7D  0E02  2918          MOVQ    CLMPRO$Q_CURTIME(R2), - ; Save time from message
```

```
                                0E08  2919                      CLUB$Q_NEWTIME(R4)
009C C4   00000000'GF   7D      0E08  2920              MOVQ    G^EXE$GQ_SYSTIME, -       ; Save current system time
                                0E11  2921                      CLUB$Q_NEWTIME_REF(R4)    ;   as a reference
   26 A4   31 A2   06   28      0E11  2922              MOVC3   #CLUB$S_FSYSID, -         ; Cluster founding system ID
                                0E17  2923                      CLMPRO$B_FSYSID(R2), -
                                0E17  2924                      CLUB$B_FSYSID(R4)
                        52   6E 7D      0E17  2925 30$:     MOVQ    (SP),R2                   ; Restore registers
                     54   64 A3 D0      0E1A  2926          MOVL    CSB$L_CLUB(R3),R4
                 46 A4   1A A2 B0      0E1E  2927          MOVW    CLMPRO$W_QDVOTES(R2), - ; Quorum disk votes
                                0E23  2928                      CLUB$W_NEWQDVOTES(R4)
              00A4 C4   16 A2 B0      0E23  2929          MOVW    CLMPRO$W_QUORUM(R2), -  ; Cluster quorum
                                0E29  2930                      CLUB$W_NEWQUORUM(R4)
            1C A4   04000000 8F CA      0E29  2931          BICL    #CLUB$M_QF_NEWVOTE, -   ; Assume no quorum disk membership
                                0E31  2932                      CLUB$L_FLAGS(R4)
                  0C 30 A2   00 E1      0E31  2933          BBC     #CLMPRO$V_QF_VOTE, -    ; Branch if no quorum disk membership
                                0E36  2934                      CLMPRO$B_FLAGS(R2),35$
                        51   D4 0E36  2935          CLRL    R1                        ; Assume failure
                  42 1C A4   01 E1      0E38  2936          BBC     #CLUB$V_QF_ACTIVE, -    ; Branch if disk not locally active
                                0E3D  2937                      CLUB$L_FLAGS(R4),60$      ;   and reject proposal
                  00 1C A4   1A E2      0E3D  2938          BBSS    #CLUB$V_QF_NEWVOTE, -   ; Set quorum disk membership
                                0E42  2939                      CLUB$L_FLAGS(R4),35$
 00EC C4   37 A2   20   28      0E42  2940 35$:     MOVC3   #CLUB$S_NODEMAP, -        ; Save map of cluster nodes
                                0E49  2941                      CLMPRO$B_NODEMAP(R2), - ;   in CLUB
                                0E49  2942                      CLUB$B_NODEMAP(R4)
                        52   6E D0      0E49  2943          MOVL    (SP),R2                   ; Restore message address
                        51   01 D0      0E4C  2944          MOVL    #1,R1                     ; Anticipate success
                           F1AE' 30      0E4F  2945          BSBW    CNX$SCAN_CSBS             ; Iterate over all CSBs
                        2A 50 E9      0E52  2946          BLBC    R0,60$                    ; Branch when done
 60 A3   00020002 8F D3      0E55  2947          BITL    #CSB$M_SELECTED ! -       ; Check for SELECTED or
                                0E5D  2948                       CSB$M_MEMBER, -          ;   MEMBER
                                0E5D  2949                      CSB$L_STATUS(R3)
                     1F 13      0E5D  2950          BEQL    50$                       ; Branch and ignore if neither
 60 A3   00020000 8F C8      0E5F  2951          BISL2   #CSB$M_SELECTED, -        ; Mark the node selected
                                0E67  2952                      CSB$L_STATUS(R3)
                  50   4C A3 3C      0E67  2953          MOVZWL  CSB$W_CSID_IDX(R3),R0     ; CSID index
               0A 37 A2   50 E5      0E6B  2954          BBCC    R0, -                     ; Branch if node is not in proposed
                                0E70  2955                      CLMPRO$B_NODEMAP(R2), - ;   cluster
                                0E70  2956                      40$
               09 60 A3   00 E1      0E70  2957          BBC     #CSB$V_LONG_BREAK, -      ; Branch if no long break seen
                                0E75  2958                      CSB$L_STATUS(R3),50$
                        51   D4 0E75  2959          CLRL    R1                        ; Flag failure
                     F186' 31      0E77  2960          BRW     CNX$SCAN_CSBS_EXIT        ; Exit from CSB scanning loop
                                0E7A  2961
                                0E7A  2962 40$:     BUG_CHECK       CNXMGRERR,FATAL ; Consistency check
                                0E7E  2963
                        05   0E7E  2964 50$:     RSB
                                0E7F  2965
                  0C AE   51 D0      0E7F  2966 60$:     MOVL    R1,12(SP)                 ; Save status
                     07 51 E9      0E83  2967          BLBC    R1,70$                    ; Branch if NAK required
               37 A2   20   00 3B      0E86  2968          SKPC    #0,#CLMPRO$S_NODEMAP, - ; Look for any missing nodes
                                0E8B  2969                      CLMPRO$B_NODEMAP(R2)
                        ED 12      0E8B  2970          BNEQ    40$                       ; Some node(s) is not accounted for
                  0A 0C AE   E9      0E8D  2971 70$:     BLBC    12(SP),80$                ; Branch on NAK pending
                     F16C' 30      0E91  2972          BSBW    CNX$DIRVEC_ADJ            ; Adjust size of lock manager directory vect
                  04 50 E8      0E94  2973          BLBS    R0,80$                    ; Branch on success
                  0C AE   50 D0      0E97  2974          MOVL    R0,12(SP)                 ; Make sure proposal is NAKed
                        2C BA 0E9B  2975 80$:     POPR    #^M<R2,R3,R5>             ; Restore needed registers
```

```
                                                I  2
                50   09    9A  0E9D  2976          MOVZBL  #CLMCNX$K_FNC_JOIN,R0    ; Facility specific function code
                     0466  30  0E A0  2977          BSBW    CNX$INIT_STD_RESP        ; Init CDRP for standard response
             36 A5    8E    F6  0E A3  2978          CVTLB   (SP)+,CDRP$L_VAL3+2(R5)  ; Store success/failure flag
                 F156'     31  0E A7  2979          BRW     CNX$RESP_FORGET          ; Queue response message
```

```
                                0EAA  2981          .SBTTL  CNX$RCVD_QUORUM  - Quorum Update Message Received
                                0EAA  2982
                                0EAA  2983 ;++
                                0EAA  2984 ;
                                0EAA  2985 ; FUNCTIONAL DESCRIPTION:
                                0EAA  2986 ;
                                0EAA  2987 ;       This routine is called when a quorum update request message is received.
                                0EAA  2988 ;       This is a Phase 1 message.  The validity of the message is checked and
                                0EAA  2989 ;       the new quorum value is saved in a temporary location.
                                0EAA  2990 ;
                                0EAA  2991 ; CALLING SEQUENCE:
                                0EAA  2992 ;
                                0EAA  2993 ;       JSB     CNX$RCVD_QUORUM
                                0EAA  2994 ;       IPL is IPL$_SCS
                                0EAA  2995 ;
                                0EAA  2996 ; INPUT PARAMETERS:
                                0EAA  2997 ;
                                0EAA  2998 ;       R2:     Message address
                                0EAA  2999 ;       R3:     CSB of sending system
                                0EAA  3000 ;       R4:     PDT address
                                0EAA  3001 ;       R5:     CDRP address (uninitialized)
                                0EAA  3002 ;
                                0EAA  3003 ; OUTPUT PARAMETERS:
                                0EAA  3004 ;
                                0EAA  3005 ;       NONE
                                0EAA  3006 ;
                                0EAA  3007 ; COMPLETION CODES:
                                0EAA  3008 ;
                                0EAA  3009 ;       NONE
                                0EAA  3010 ;
                                0EAA  3011 ; SIDE EFFECTS:
                                0EAA  3012 ;
                                0EAA  3013 ;       R0-R5 may be destroyed.
                                0EAA  3014 ;--
                                0EAA  3015
                                0EAA  3016 CNX$RCVD_QUORUM::
                    2C    BB    0EAA  3017          PUSHR   #^M<R2,R3,R5>           ; Save needed registers
                F151'     30    0EAC  3018          BSBW    CNX$INIT_CDRP           ; Initialize the CDRP for the response
            52    6E    7D    0EAF  3019          MOVQ    (SP),R2                 ; Restore message address and CSB address
               0494     30    0EB2  3020          BSBW    MSG_CHECK               ; Validate message
               F148'    30    0EB5  3021          BSBW    CNX$SCAN_CSBS           ; Iterate over all CSBs
              0B 50     E9    0EB8  3022          BLBC    R0,20$                  ; Branch when done
      05 60 A3   01     E1    0EBB  3023          BBC     #CSB$V_MEMBER, -        ; Branch if not a member node
                                0EC0  3024                  CSB$L_STATUS(R3),10$
      00 60 A3   11     E2    0EC0  3025          BBSS    #CSB$V_SELECTED, -      ; Select each menber
                                0EC5  3026                  CSB$L_STATUS(R3),10$
                   05     05    0EC5  3027 10$:     RSB
                                0EC6  3028
            52    6E    7D    0EC6  3029 20$:     MOVQ    (SP),R2                 ; Restore message buffer address
    00A4 C4    16 A2    B0    0EC9  3030          MOVW    CLMPRO$W_QUORUM(R2), -  ; Temporarily save new quorum value
                                0ECF  3031                  CLUB$W_NEWQUORUM(R4)
      46 A4    1A A2    B0    0ECF  3032          MOVW    CLMPRO$W_QDVOTES(R2), - ; Quorum disk votes
                                0ED4  3033                  CLUB$W_NEWQDVOTES(R4)
1C A4    04000000 8F    CA    0ED4  3034          BICL    #CLUB$M_QF_NEWVOTE, -   ; Assume no quorum disk membership
                                0EDC  3035                  CLUB$L_FLAGS(R4)
            51    01    D0    0EDC  3036          MOVL    #1,R1                   ; Assume that we will accept proposal
      0F 30 A2    00    E1    0EDF  3037          BBC     #CLMPRO$V_QF_VOTE, -    ; Branch if no quorum disk membership
```

```
                                OEE4  3038                    CLMPRO$B_FLAGS(R2),30$
                          51    D4  OEE4  3039          CLRL  R1                      ; Assume failure
                    08 1C A4    01  E1  OEE6  3040          BBC   #CLUB$V_QF_ACTIVE, -  ; Branch if disk not locally active
                                OEEB  3041                    CLUB$L_FLAGS(R4),30$  ;   and reject proposal
                          51    01  D0  OEEB  3042          MOVL  #1,R1                 ; Accept proposal
                    00 1C A4    1A  E2  OEEE  3043          BBSS  #CLUB$V_QF_NEWVOTE, - ; Set quorum disk membership
                                OEF3  3044                    CLUB$L_FLAGS(R4),30$
                          2C    BA  OEF3  3045 30$:       POPR  #^M<R2,R3,R5>         ; Restore needed registers
                          51    DD  OEF5  3046          PUSHL R1                    ; Save status
                       50 0D    9A  OEF7  3047          MOVZBL #CLMCNX$K_FNC_QUORUM,R0 ; Facility specific function code
                          040C  30  OEFA  3048          BSBW  CNX$INIT_STD_RESP     ; Init CDRP for standard response
                    36 A5 8E    F6  OEFD  3049          CVTLB (SP)+,CDRP$L_VAL3+2(R5) ; Store success flag
                          FOFC' 31  OF01  3050          BRW   CNX$RESP_FORGET       ; Send response message and forget it
```

```
                                0F04  3052          .SBTTL  BLD_VEC_MSG - Build Message Describing Vector Slot
                                0F04  3053
                                0F04  3054 ;++
                                0F04  3055 ;
                                0F04  3056 ; FUNCTIONAL DESCRIPTION:
                                0F04  3057 ;
                                0F04  3058 ;       This routine builds a message describing an empty cluster vector slot.
                                0F04  3059 ;       This is needed in order to guarantee consistent, cluster-wide assignment
                                0F04  3060 ;       of unique CSIDs.
                                0F04  3061 ;
                                0F04  3062 ; CALLING SEQUENCE:
                                0F04  3063 ;
                                0F04  3064 ;       JSB     BLD_VEC_MSG
                                0F04  3065 ;       IPL is IPL$_SCS
                                0F04  3066 ;
                                0F04  3067 ; INPUT PARAMETERS:
                                0F04  3068 ;
                                0F04  3069 ;       R2:     Address of message buffer
                                0F04  3070 ;       R3:     Address of CSB
                                0F04  3071 ;       R4:     Address of PDT
                                0F04  3072 ;       R5:     Address of CDRP
                                0F04  3073 ;               CDRP$L_VAL1(R5): Byte 0 contains facility code (negated for response
                                0F04  3074 ;                                Byte 1 contains facility specific function code
                                0F04  3075 ;               CDRP$L_VAL2(R5): Transition ID (from CLUB$L_CUR_XTN)
                                0F04  3076 ;               CDRP$L_VAL3(R5): Byte 0 contains transition phase (from CLUB$B_CUR_P
                                0F04  3077 ;                                Byte 1 contains transition code (from CLUB$B_XTN_CO
                                0F04  3078 ;                                Byte 2 contains success/failure flag
                                0F04  3079 ;                                Byte 3 contains reply code
                                0F04  3080 ;
                                0F04  3081 ; OUTPUT PARAMETERS:
                                0F04  3082 ;
                                0F04  3083 ;       NONE
                                0F04  3084 ;
                                0F04  3085 ; COMPLETION CODES:
                                0F04  3086 ;
                                0F04  3087 ;       NONE
                                0F04  3088 ;
                                0F04  3089 ; SIDE EFFECTS:
                                0F04  3090 ;
                                0F04  3091 ;       R0 and R1 are destroyed
                                0F04  3092 ;
                                0F04  3093 ;--
                                0F04  3094
                                0F04  3095 BLD_VEC_MSG:
            0432    30          0F04  3096          BSBW    BLD_STD_MSG                 ; Fill in standard fields
      50    48 A5   D0          0F07  3097          MOVL    CDRP$L_VAL8(R5),R0          ; Slot index
      14 A2   50    F7          0F0B  3098          CVTLW   R0,CLMVEC$W_INDEX(R2)       ; Store slot index
51  00000000'GF    D0          0F0F  3099          MOVL    G^CLU$GL_CLUSVEC,R1         ; Address of cluster vector
      50    6140    D0          0F16  3100          MOVL    (R1)[R0],R0                 ; Get contents of slot
            07      19          0F1A  3101          BLSS    10$                         ; Branch if slot in use
      16 A2   50    F7          0F1C  3102          CVTLW   R0,CLMVEC$W_SEQUENCE(R2)    ; Last sequence number used
            01      1D          0F20  3103          BVS     10$                         ; Branch on overflow
                    05          0F22  3104          RSB
                                0F23  3105
                                0F23  3106 10$:     BUG_CHECK       CNXMGRERR,FATAL ; Consistency check
```

```
                        0F27  3108          .SBTTL  CNX$RCVD_VEC - Cluster vector slot information received
                        0F27  3109
                        0F27  3110 ;++
                        0F27  3111 ;
                        0F27  3112 ; FUNCTIONAL DESCRIPTION:
                        0F27  3113 ;
                        0F27  3114 ;       This routine is called when a cluster vector slot description
                        0F27  3115 ;       is received.
                        0F27  3116 ;       No response or immediate acknowledgement is sent.
                        0F27  3117 ;
                        0F27  3118 ; CALLING SEQUENCE:
                        0F27  3119 ;
                        0F27  3120 ;       JSB     CNX$RCVD_VEC
                        0F27  3121 ;       IPL is IPL$_SCS
                        0F27  3122 ;
                        0F27  3123 ; INPUT PARAMETERS:
                        0F27  3124 ;
                        0F27  3125 ;       R2:     Message address
                        0F27  3126 ;       R3:     CSB of sending system
                        0F27  3127 ;       R4:     PDT address
                        0F27  3128 ;
                        0F27  3129 ; OUTPUT PARAMETERS:
                        0F27  3130 ;
                        0F27  3131 ;       NONE
                        0F27  3132 ;
                        0F27  3133 ; COMPLETION CODES:
                        0F27  3134 ;
                        0F27  3135 ;       NONE
                        0F27  3136 ;
                        0F27  3137 ; SIDE EFFECTS:
                        0F27  3138 ;
                        0F27  3139 ;       R0-R5 may be destroyed.
                        0F27  3140 ;--
                        0F27  3141
                        0F27  3142 CNX$RCVD_VEC::
            041F    30  0F27  3143          BSBW    MSG_CHECK                       ; Validate message
50  00000000'GF    D0  0F2A  3144          MOVL    G^CLU$GL_CLUSVEC,R0             ; Address of cluster vector
      51   14 A2   3C  0F31  3145          MOVZWL  CLMVEC$W_INDEX(R2),R1           ; Slot index
00000000'GF   51    B1  0F35  3146          CMPW    R1,G^CLU$GW_MAXINDEX            ; Valid index?
            09    1E  0F3C  3147          BGEQU   10$                             ; Invalid index
    6041   16 A2   3C  0F3E  3148          MOVZWL  CLMVEC$W_SEQUENCE(R2), -        ; Store last used sequence number
                       0F43  3149                  (R0)[R1]
            F0BA'   30  0F43  3150          BSBW    CNX$DEALL_MSG_BUF_CSB           ; Deallocate message buffer
                05  0F46  3151          RSB                                     ; Return
                       0F47  3152
                       0F47  3153 10$:      BUG_CHECK       CNXMGRERR,FATAL ; Consistency check
```

```
                                0F4B  3155          .SBTTL  DESCRIBE_NODE - Describe one node to another node
                                0F4B  3156
                                0F4B  3157 ;++
                                0F4B  3158 ;
                                0F4B  3159 ; FUNCTIONAL DESCRIPTION:
                                0F4B  3160 ;
                                0F4B  3161 ;       This routine tells one node about another node.  It returns
                                0F4B  3162 ;       asynchronously with information indicating that the first
                                0F4B  3163 ;       node can or cannot see the second node.
                                0F4B  3164 ;
                                0F4B  3165 ; CALLING SEQUENCE:
                                0F4B  3166 ;
                                0F4B  3167 ;       JSB     DESCRIBE_NODE
                                0F4B  3168 ;       IPL is IPL$_SCS
                                0F4B  3169 ;
                                0F4B  3170 ; INPUT PARAMETERS:
                                0F4B  3171 ;
                                0F4B  3172 ;       R5 is address of a CDRP
                                0F4B  3173 ;       R3 is CSB address of node to talk to
                                0F4B  3174 ;       R2 is CSB address of node to describe
                                0F4B  3175 ;
                                0F4B  3176 ; OUTPUT PARAMETERS:
                                0F4B  3177 ;
                                0F4B  3178 ;       R5 is address of a CDRP
                                0F4B  3179 ;       R3 is CSB address of node talked to
                                0F4B  3180 ;       R2 is CSB address of node described
                                0F4B  3181 ;
                                0F4B  3182 ; COMPLETION CODES:
                                0F4B  3183 ;
                                0F4B  3184 ;       R0 indicates that a response was received from
                                0F4B  3185 ;             the node vs. connection failed
                                0F4B  3186 ;       R1 is status received from the node.
                                0F4B  3187 ;
                                0F4B  3188 ; SIDE EFFECTS:
                                0F4B  3189 ;
                                0F4B  3190 ;       CDRP$L_VAL1-CDRP$L_VAL5 are used for this routine's context.
                                0F4B  3191 ;
                                0F4B  3192 ;--
                                0F4B  3193
                                0F4B  3194 DESCRIBE_NODE:
          F0B2'   30            0F4B  3195          BSBW    CNX$INIT_CDRP           ; Initialize the CDRP
    54    64 A3   D0            0F4E  3196          MOVL    CSB$L_CLUB(R3),R4       ; Address of CLUB
 38 A5     52   D0              0F52  3197          MOVL    R2,CDRP$L_VAL4(R5)      ; Node to describe
       3C A5 8ED0               0F56  3198          POPL    CDRP$L_VAL5(R5)         ; Return address
    50    05   9A               0F5A  3199          MOVZBL  #CLMCNX$K_FNC_DESC,R0   ; Message code
          03B2   30              0F5D  3200          BSBW    INIT_STD_MSG            ; Do standard message initialization
 20 A5    01   D0              0F60  3201          MOVL    #1,CDRP$L_RSPID(R5)     ; Indicate that response is needed
4C A5   87'AF   9E              0F64  3202          MOVAB   B^BLD_DESC_MSG, -       ; Address of routine to build
                                0F69  3203                  CDRP$L_MSGBLD(R5)       ;  status message
          F094'   30            0F69  3204          BSBW    CNX$SEND_MSG_CSB        ; Send message
                                0F6C  3205 ;
                                0F6C  3206 ;       We are resumed here when the response message arrives.
                                0F6C  3207 ;       Registers contain:
                                0F6C  3208 ;               R0:     Status
                                0F6C  3209 ;               R2:     Address of message buffer
                                0F6C  3210 ;               R3:     Address of CSB
                                0F6C  3211 ;               R4:     Address of PDT
```

```
                                                B 3
                     0F6C  3212 ;                       R5:     Address of CDRP
                     0F6C  3213 ;
          11 50      E9  0F6C  3214                BLBC    R0,10$                          ; Error sending message
     7E   12 A2      9A  0F6F  3215                MOVZBL  CLMCNX$B_ACK(R2),-(SP)          ; Response from remote node
          50         DD  0F73  3216                PUSHL   R0                              ; Save return status
           F088'     30  0F75  3217                BSBW    CNX$DEALL_MSG_BUF_CSB           ; Release message buffer
                     0F78  3218                    DEALLOC_RSPID                           ; Deallocate response ID
          03         BA  0F7E  3219                POPR    #^M<R0,R1>                      ; Restore status and return ACK/NAK
     52   38 A5      D0  0F80  3220 10$:           MOVL    CDRP$L_VAL4(R5),R2              ; Address of described CSB
          3C B5      17  0F84  3221                JMP     @CDRP$L_VAL5(R5)                ; Return address
```

```
                        0F87  3223          .SBTTL  BLD_DESC_MSG - Build Message describing a node
                        0F87  3224
                        0F87  3225 ;++
                        0F87  3226 ;
                        0F87  3227 ; FUNCTIONAL DESCRIPTION:
                        0F87  3228 ;
                        0F87  3229 ;       This routine builds a message describing a node.  It uses the
                        0F87  3230 ;       standard message building routine to build the message header.
                        0F87  3231 ;
                        0F87  3232 ; CALLING SEQUENCE:
                        0F87  3233 ;
                        0F87  3234 ;       JSB     BLD_DESC_MSG
                        0F87  3235 ;       IPL is IPL$_SCS
                        0F87  3236 ;
                        0F87  3237 ; INPUT PARAMETERS:
                        0F87  3238 ;
                        0F87  3239 ;       R2:     Address of message buffer
                        0F87  3240 ;       R3:     Address of CSB
                        0F87  3241 ;       R4:     Address of FDT
                        0F87  3242 ;       R5:     Address of CDRP
                        0F87  3243 ;               CDRP$L_VAL1(R5): Byte 0 contains facility code (negated for response
                        0F87  3244 ;                                Byte 1 contains facility specific function code
                        0F87  3245 ;               CDRP$L_VAL2(R5): Transition ID (from CLUB$L_CUR_XTN)
                        0F87  3246 ;               CDRP$L_VAL3(R5): Byte 0 contains transition phase (from CLUB$B_CUR_P
                        0F87  3247 ;                                Byte 1 contains transition code (from CLUB$B_XTN_CO
                        0F87  3248 ;                                Byte 2 contains success/failure flag
                        0F87  3249 ;                                Byte 3 contains reply code
                        0F87  3250 ;               CDRP$L_VAL4(R5): Address of CSB to describe
                        0F87  3251 ;
                        0F87  3252 ; OUTPUT PARAMETERS:
                        0F87  3253 ;
                        0F87  3254 ;       NONE
                        0F87  3255 ;
                        0F87  3256 ; COMPLETION CODES:
                        0F87  3257 ;
                        0F87  3258 ;       NONE
                        0F87  3259 ;
                        0F87  3260 ; SIDE EFFECTS:
                        0F87  3261 ;
                        0F87  3262 ;       R0 and R1 are destroyed
                        0F87  3263 ;
                        0F87  3264 ;--
                        0F87  3265
                        0F87  3266 BLD_DESC_MSG:
                 03AF   30   0F87  3267          BSBW    BLD_STD_MSG                     ; Fill in standard fields
          50    38 A5  D0   0F8A  3268          MOVL    CDRP$L_VAL4(R5),R0              ; Address of CSB to describe
          51    68 A0  D0   0F8E  3269          MOVL    CSB$L_SB(R0),R1                 ; Address of SB of described system
             3F         BB   0F92  3270          PUSHR   #^M<R0,R1,R2,R3,R4,R5>          ; Save registers
14 A2     18 A1   06   28   0F94  3271          MOVC3   #CLMNOD$S_SYSTEMID, -           ; Fill in described system ID
                               0F9A  3272                  SB$B_SYSTEMID(R1), -
                               0F9A  3273                  CLMNOD$B_SYSTEMID(R2)
             3F         BA   0F9A  3274          POPR    #^M<R0,R1,R2,R3,R4,R5>          ; Restore registers
      1C A2  2C A1   7D   0F9C  3275          MOVQ    SB$Q_SWINCARN(R1), -            ; Fill in described system
                               0FA1  3276                  CLMNOD$Q_SWINCARN(R2)       ;   software incarnation number
      24 A2  4C A0   D0   0FA1  3277          MOVL    CSB$L_CSID(R0), -               ; Described system CSID
                               0FA6  3278                  CLMNOD$L_CSID(R2)
                         05   0FA6  3279          RSB
```

```
                        0FA7  3281          .SBTTL  CNX$RCVD_DESC  - Node description receiv
                        0FA7  3282
                        0FA7  3283 ;++
                        0FA7  3284 ;
                        0FA7  3285 ; FUNCTIONAL DESCRIPTION:
                        0FA7  3286 ;
                        0FA7  3287 ;       This routine is called when a node description message is received.
                        0FA7  3288 ;       A response must be sent which ACKs or NAKs the request.  The
                        0FA7  3289 ;       request is NAKd if this node does not have a connection.
                        0FA7  3290 ;
                        0FA7  3291 ; CALLING SEQUENCE:
                        0FA7  3292 ;
                        0FA7  3293 ;       JSB     CNX$RCVD_DESC
                        0FA7  3294 ;       IPL is IPL$_SCS
                        0FA7  3295 ;
                        0FA7  3296 ; INPUT PARAMETERS:
                        0FA7  3297 ;
                        0FA7  3298 ;       R2:     Message address
                        0FA7  3299 ;       R3:     CSB of sending system
                        0FA7  3300 ;       R4:     PDT address
                        0FA7  3301 ;       R5:     CDRP address (uninitialized)
                        0FA7  3302 ;
                        0FA7  3303 ; OUTPUT PARAMETERS:
                        0FA7  3304 ;
                        0FA7  3305 ;       NONE
                        0FA7  3306 ;
                        0FA7  3307 ; COMPLETION CODES:
                        0FA7  3308 ;
                        0FA7  3309 ;       NONE
                        0FA7  3310 ;
                        0FA7  3311 ; SIDE EFFECTS:
                        0FA7  3312 ;
                        0FA7  3313 ;       R0-R5 may be destroyed.
                        0FA7  3314 ;--
                        0FA7  3315
                        0FA7  3316 CNX$RCVD_DESC::
                  2C    BB  0FA7  3317          PUSHR   #^M<R2,R3,R5>           ; Save needed registers
              F054'     30  0FA9  3318          BSBW    CNX$INIT_CDRP           ; Initialize the CDRP for the response
           52   6E     D0  0FAC  3319          MOVL    (SP),R2                 ; Restore message address
              0397     30  0FAF  3320          BSBW    MSG_CHECK               ; Validate message
                55     D4  0FB2  3321          CLRL    R5                      ; Result CSB address
              F049'     30  0FB4  3322          BSBW    CNX$SCAN_CSBS           ; Iterate over all CSBs
             27 50      E9  0FB7  3323          BLBC    R0,5$                   ; Branch when done
        50   68 A3      D0  0FBA  3324          MOVL    CSB$L_SB(R3),R0         ; Address of System Block
                0C      BB  0FBE  3325          PUSHR   #^M<R2,R3>              ; Save message buffer and CSB address
14 A2   18 A0   06      29  0FC0  3326          CMPC3   #SB$S_SYSTEMID, -       ; Compare System IDs
                            0FC6  3327                  SB$B_SYSTEMID(R0), -    ;   between SB and
                            0FC6  3328                  CLMNOD$B_SYSTEMID(R2)   ;   incoming message
                0C      BA  0FC6  3329          POPR    #^M<R2,R3>              ; Restore message buffer and CSB address
                16      12  0FC8  3330          BNEQ    2$                      ; No match
                0C      BB  0FCA  3331          PUSHR   #^M<R2,R3>              ; Save message buffer and CSB address
        50   68 A3      D0  0FCC  3332          MOVL    CSB$L_SB(R3),R0         ; Address of System Block
                            0FD0  3333 ;
                            0FD0  3334 ; Use SWINCARN out of SB because local node's CSB doesn't contain a valid
                            0FD0  3335 ; SWINCARN -- it was created and intialized before SWINCARN for the local node
                            0FD0  3336 ; became meaningful.
                            0FD0  3337 ;
```

```
                1C A2    2C A0    08    29  0FD0  3338          CMPC3   #SB$S_SWINCARN, -       ; Compare software incarnation number
                                           0FD6  3339                  SB$Q_SWINCARN(R0), -    ;   between CSB and
                                           0FD6  3340                  CLMNOD$Q_SWINCARN(R2)   ;   incoming message
                                     0C    BA  0FD6  3341          POPR    #^M<R2,R3>              ; Restore message buffer and CSB address
                                     06    12  0FD8  3342          BNEQ    2$                      ; No match
                           55        53    D0  0FDA  3343          MOVL    R3,R5                   ; Store result
                              F020'  31  0FDD  3344          BRW     CNX$SCAN_CSBS_EXIT      ; Exit iteration with result
                                           0FE0  3345
                                     05    0FE0  3346 2$:       RSB                             ; Return and iterate
                                           0FE1  3347
                           53        55    D0  0FE1  3348 5$:       MOVL    R5,R3                   ; CSB address of described node
                                     29    13  0FE4  3349          BEQL    25$                     ; Described node not found
                                     18    E0  0FE6  3350          BBS     #CSB$V_LOCAL,-          ; Branch if local CSB
                                 0A 60 A3        0FE8  3351                  CSB$L_STATUS(R3),10$
                        17 60 A3       19    E1  0FEB  3352          BBC     #CSB$V_STATUS_RCVD, -   ; Branch if no status received
                                                0FF0  3353                  CSB$L_STATUS(R3),20$
                        12 60 A3       00    E0  0FF0  3354          BBS     #CSB$V_LONG_BREAK, -    ; Branch if long break seen
                                                0FF5  3355                  CSB$L_STATUS(R3),20$    ;   and NAK request
                60 A3    00020000 8F    C8  0FF5  3356 10$:      BISL2   #CSB$M_SELECTED, -      ; Mark node SELECTED
                                                0FFD  3357                  CSB$L_STATUS(R3)
                        4C A3    24 A2  D0  0FFD  3358          MOVL    CLMNOD$L_CSID(R2), -    ; Save node identification
                                                1002  3359                  CSB$L_CSID(R3)          ;   for described node
                                 50    01  D0  1002  3360          MOVL    #1,R0                   ; ACK node description
                                       0A  11  1005  3361          BRB     30$
                                           1007  3362 ;
                                           1007  3363 ; The described node is unknown or no valid connection exists.
                                           1007  3364 ; Return a NAK.
                                           1007  3365 ;
                60 A3    00020000 8F    CA  1007  3366 20$:      BICL2   #CSB$M_SELECTED, -      ; Mark node deSELECTED
                                                100F  3367                  CSB$L_STATUS(R3)
                                 50    D4  100F  3368 25$:      CLRL    R0                      ; NAK node description
                                 2C    BA  1011  3369 30$:      POPR    #^M<R2,R3,R5>           ; Restore needed registers
                                 50    DD  1013  3370          PUSHL   R0                      ; Save status
                              50 05    9A  1015  3371          MOVZBL  #CLMCNX$K_FNC_DESC,R0   ; Facility specific function code
                                 02EE  30  1018  3372          BSBW    CNX$INIT_STD_RESP       ; Init CDRP for standard response
                           36 A5 8E    F6  101B  3373          CVTLB   (SP)+,CDRP$L_VAL3+2(R5) ; Store success/failure flag
                                 EFDE' 31  101F  3374          BRW     CNX$RESP_FORGET         ; Send response message and forget it
```

```
                        1022  3376          .SBTTL  BLD_TOPOLOGY_MSG - Build Message Describing Cluster Topology
                        1022  3377
                        1022  3378 ;++
                        1022  3379 ;
                        1022  3380 ; FUNCTIONAL DESCRIPTION:
                        1022  3381 ;
                        1022  3382 ;       This routine builds a message describing the connectivity of the cluster
                        1022  3383 ;       as known to the local node.  The description is optimistic in that some
                        1022  3384 ;       connections that are reported as good may in fact be bad.  The oppositve
                        1022  3385 ;       cannot occur.  Information about the quorum disk is not passed in this
                        1022  3386 ;       message.
                        1022  3387 ;
                        1022  3388 ; CALLING SEQUENCE:
                        1022  3389 ;
                        1022  3390 ;       JSB     BLD_TOPOLOGY_MSG
                        1022  3391 ;       IPL is IPL$_SCS
                        1022  3392 ;
                        1022  3393 ; INPUT PARAMETERS:
                        1022  3394 ;
                        1022  3395 ;       R2:     Address of message buffer
                        1022  3396 ;       R3:     Address of CSB
                        1022  3397 ;       R4:     Address of PDT
                        1022  3398 ;       R5:     Address of CDRP
                        1022  3399 ;               CDRP$L_VAL1(R5): Byte 0 contains facility code (negated for response
                        1022  3400 ;                                Byte 1 contains facility specific function code
                        1022  3401 ;               CDRP$L_VAL2(R5): Transition ID (from CLUB$L_CUR_XTN)
                        1022  3402 ;               CDRP$L_VAL3(R5): Byte 0 contains transition phase (from CLUB$B_CUR_P
                        1022  3403 ;                                Byte 1 contains transition code (from CLUB$B_XTN_CO
                        1022  3404 ;                                Byte 2 contains success/failure flag
                        1022  3405 ;                                Byte 3 contains reply code
                        1022  3406 ;
                        1022  3407 ; OUTPUT PARAMETERS:
                        1022  3408 ;
                        1022  3409 ;       NONE
                        1022  3410 ;
                        1022  3411 ; COMPLETION CODES:
                        1022  3412 ;
                        1022  3413 ;       NONE
                        1022  3414 ;
                        1022  3415 ; SIDE EFFECTS:
                        1022  3416 ;
                        1022  3417 ;       R0 and R1 are destroyed
                        1022  3418 ;
                        1022  3419 ;--
                        1022  3420
                        1022  3421 BLD_TOPOLOGY_MSG:
                  0314   30  1022  3422          BSBW    BLD_STD_MSG                     ; Fill in standard fields
           50    64 A3   D0  1025  3423          MOVL    CSB$L_CLUB(R3),R0               ; Get CLUB address
           51    10 A0   D0  1029  3424          MOVL    CLUB$L_LOCAL_CSB(R0),R1         ; Local CSB Address
              3C   BB  102D  3425          PUSHR   #^M<R2,R3,R4,R5>                ; Save registers
14 A2  008C C1    20   28  102F  3426          MOVC3   #CLMTOP$S_NODEMAP, -            ; Copy data into message buffer
                          1036  3427                  CSB$B_NODEMAP(R1), -
                          1036  3428                  CLMTOP$B_NODEMAP(R2)
              3C   BA  1036  3429          POPR    #^M<R2,R3,R4,R5>                ; Restore registers
                   05  1038  3430          RSB                                     ; Return and iterate
```

```
                                1039  3432          .SBTTL  CNX$RCVD_TOPOLOGY  - Topology Request Message Received
                                1039  3433
                                1039  3434 ;++
                                1039  3435 ;
                                1039  3436 ; FUNCTIONAL DESCRIPTION:
                                1039  3437 ;
                                1039  3438 ;       This routine is called when a topology request message is received.
                                1039  3439 ;       A response must be sent describes the connectivity seen by the
                                1039  3440 ;       receiving node.  The message itself is used to update the receiving
                                1039  3441 ;       nodes' understanding of the sending nodes' connectivity.
                                1039  3442 ;
                                1039  3443 ; CALLING SEQUENCE:
                                1039  3444 ;
                                1039  3445 ;       JSB     CNX$RCVD_TOPOLOGY
                                1039  3446 ;       IPL is IPL$_SCS
                                1039  3447 ;
                                1039  3448 ; INPUT PARAMETERS:
                                1039  3449 ;
                                1039  3450 ;       R2:     Message address
                                1039  3451 ;       R3:     CSB of sending system
                                1039  3452 ;       R4:     PDT address
                                1039  3453 ;       R5:     CDRP address (uninitialized)
                                1039  3454 ;
                                1039  3455 ; OUTPUT PARAMETERS:
                                1039  3456 ;
                                1039  3457 ;       NONE
                                1039  3458 ;
                                1039  3459 ; COMPLETION CODES:
                                1039  3460 ;
                                1039  3461 ;       NONE
                                1039  3462 ;
                                1039  3463 ; SIDE EFFECTS:
                                1039  3464 ;
                                1039  3465 ;       R0-R5 may be destroyed.
                                1039  3466 ;--
                                1039  3467
                                1039  3468 CNX$RCVD_TOPOLOGY::
                     2C   BB    1039  3469          PUSHR   #^M<R2,R3,R5>           ; Save needed registers
                  EFC2'   30    103B  3470          BSBW    CNX$INIT_CDRP           ; Initialize the CDRP for the response
              52    6E   7D    103E  3471          MOVQ    (SP),R2                 ; Restore message address and CSB address
                0305   30    1041  3472          BSBW    MSG_CHECK               ; Validate message
008C C3   14 A2     20   28    1044  3473          MOVC3   #CSB$S_NODEMAP, -       ; Fill in connectivity in sending node's
                                104B  3474                  CLMTOP$B_NODEMAP(R2), - ;   CSB
                                104B  3475                  CSB$B_NODEMAP(R3)
                     2C   BA    104B  3476          POPR    #^M<R2,R3,R5>           ; Restore needed registers
              54    64 A3   D0    104D  3477          MOVL    CSB$L_CLUB(R3),R4       ; Fetch CLUB address
                 50    0F   9A    1051  3478          MOVZBL  #CLMCRX$K_FNC_TOPOLOGY,R0       ; Facility specific function code
                   02B2   30    1054  3479          BSBW    CNX$INIT_STD_RESP       ; Init CDRP for standard response
           4C A5    C8 AF   9E    1057  3480          MOVAB   B^BLD_TOPOLOGY_MSG, -   ; Store message build routine
                                105C  3481                  CDRP$L_MSGBLD(R5)
                  EFA1'   31    105C  3482          BRW     CNX$RESP_FORGET         ; Send response message and forget it
```

```
                        105F  3484          .SBTTL  SEND_PH1 - Send Phase 1 Messages
                        105F  3485
                        105F  3486 ;++
                        105F  3487 ;
                        105F  3488 ; FUNCTIONAL DESCRIPTION:
                        105F  3489 ;
                        105F  3490 ;       Send a Phase 1 message to all locked and selected
                        105F  3491 ;       nodes.  A response is required from every node.  If any response
                        105F  3492 ;       is a NAK or any connection breaks, the transition is abandoned
                        105F  3493 ;       without returning to the caller.
                        105F  3494 ;
                        105F  3495 ; CALLING SEQUENCE:
                        105F  3496 ;
                        105F  3497 ;       JSB     SEND_PH1
                        105F  3498 ;       IPL is IPL$_SCS
                        105F  3499 ;
                        105F  3500 ; INPUT PARAMETERS:
                        105F  3501 ;
                        105F  3502 ;       R4 is address of CLUB
                        105F  3503 ;               CLUB$B_CUR_CODE gives transaction type
                        105F  3504 ;               CLUB$L_CTX0 contains R0 argument for INIT_STD_MSG
                        105F  3505 ;               CLUB$L_CTX1 contains message build routine address
                        105F  3506 ;
                        105F  3507 ; OUTPUT PARAMETERS:
                        105F  3508 ;
                        105F  3509 ;       R4 is address of CLUB
                        105F  3510 ;
                        105F  3511 ; COMPLETION CODES:
                        105F  3512 ;
                        105F  3513 ;       NONE
                        105F  3514 ;
                        105F  3515 ; SIDE EFFECTS:
                        105F  3516 ;
                        105F  3517 ;       CLUB$L_RET1 is destroyed.
                        105F  3518 ;       R0-R3, R5 are destroyed.
                        105F  3519 ;--
                        105F  3520
                        105F  3521 SEND_PH1:
          74 A4 8ED0    105F  3522          POPL    CLUB$L_RET1(R4)         ; Save return address
          5A A4   B4    1063  3523          CLRW    CLUB$W_MSGCNT(R4)       ; Initialize waiting count
59 A4     40 8F   90    1066  3524          MOVB    #CLMCNX$K_PH1, -        ; Mark this as Phase 1
                        106B  3525                  CLUB$B_CUR_PHASE(R4)
           EF92'  30    106B  3526          BSBW    CNX$SCAN_CSBS           ; Iterate over all CSBs
         41 50    E9    106E  3527          BLBC    R0,40$                  ; Branch when done
1B 60 A3   11     E1    1071  3528          BBC     #CSB$V_SELECTED, -      ; Branch if not selected
                        1076  3529                  CSB$L_STATUS(R3),10$
16 60 A3   00     E0    1076  3530          BBS     #CSB$V_LONG_BREAK, -    ; Branch if permanent break
                        107B  3531                  CSB$L_STATUS(R3),10$    ;   in connection
11 60 A3   18     E0    107B  3532          BBS     #CSB$V_LOCAL, -         ; Branch if local CSB
                        1080  3533                  CSB$L_STATUS(R3),10$
           EF7D'  30    1080  3534          BSBW    CNX$ALLOC_WARMCDRP_CSB
           EF7A'  30    1083  3535          BSBW    CNX$RESOURCE_CHECK      ; Watch out for exhausted resources
         09 50    E9    1086  3536          BLBC    R0,20$                  ; No memory available
          5A A4   B6    1089  3537          INCW    CLUB$W_MSGCNT(R4)       ; Include in wait count
             07   10    108C  3538          BSBB    30$                     ; Describe nodes
           EF6F'  31    108E  3539          BRW     CNX$SCAN_CSBS_FORK      ; Fork and resume scan
                        1091  3540
```

```
                        05  1091  3541 10$:    RSB                                 ; Return and continue scan
                            1092  3542
                EF6B'   31  1093  3543 20$:    BRW     CNX$SCAN_CSBS_RETRY         ; Delay and resume scan
                            1095  3544
                            1095  3545 ;
                            1095  3546 ; Send proposal to one node
                            1095  3547 ;
          50    70 A4   D0  1095  3548 30$:    MOVL    CLUB$L_CTX0(R4),R0          ; Facility specific message code
                 0276   30  1099  3549          BSBW    INIT_STD_MSG                ; Standard CDRP message initialization
     4C A5      78 A4   D0  109C  3550          MOVL    CLUB$L_CTX1(R4), -          ; Address of routine to build
                            10A1  3551                  CDRP$L_MSGBLD(R5)           ;  status message
                EF5C'   30  10A1  3552          BSBW    CNX$SEND_MSG_CSB            ; Send message
                            10A4  3553 ;
                            10A4  3554 ;        We are resumed here when the response message arrives.
                            10A4  3555 ;        Registers contain:
                            10A4  3556 ;                R0:     Status
                            10A4  3557 ;                R2:     Address of message buffer
                            10A4  3558 ;                R3:     Address of CSB
                            10A4  3559 ;                R4:     Address of PDT
                            10A4  3560 ;                R5:     Address of CDRP
                            10A4  3561 ;
                 023F   30  10A4  3562          BSBW    CNX$PROCESS_RESPONSE        ; Deallocate storage, get status
                08 50   E8  10A7  3563          BLBS    R0,40$                      ; Branch if ACK
  1C A4   00020000 8F   C8  10AA  3564          BISL2   #CLUB$M_UNLOCK, -           ; Request that all be unlocked
                            10B2  3565                  CLUB$L_FLAGS(R4)
                            10B2  3566 ;
                            10B2  3567 ; terminate threads -- except for the last one
                            10B2  3568 ;
                5A A4   B7  10B2  3569 40$:    DECW    CLUB$W_MSGCNT(R4)
                   01   19  10B5  3570          BLSS    50$                         ; Branch when done
                        05  10B7  3571          RSB                                 ; Terminate thread
                            10B8  3572
                 0179   30  10B8  3573 50$:    BSBW    CNX$CHECK_UNLOCK            ; Check unlock request bit
              74 B4     17  10BB  3574          JMP     @CLUB$L_RET1(R4)            ; Return to caller
```

```
                                        .SBTTL  SEND_PH2 - Send a Phase 2 Notification

                                ;++
                                ;
                                ; FUNCTIONAL DESCRIPTION:
                                ;
                                ;       Send a Phase 2 notification to all locked and selected
                                ;       nodes.  All messages are sent in parallel.  Breaking
                                ;       connections are ignored.
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;       JSB     SEND_PH2
                                ;       IPL is IPL$_SCS
                                ;
                                ; INPUT PARAMETERS:
                                ;
                                ;       CLUB$B_CUR_CODE gives transaction type
                                ;
                                ; OUTPUT PARAMETERS:
                                ;
                                ;       R4 is address of CLUB
                                ;
                                ; COMPLETION CODES:
                                ;
                                ;       NONE
                                ;
                                ; SIDE EFFECTS:
                                ;
                                ;--

                                SEND_PH2:
59 A4   60 8F    90            MOVB    #CLMCNX$K_PH2, -        ; Set current phase
                                        CLUB$B_CUR_PHASE(R4)    ;   into CLUB
        EF3A'     30            BSBW    CNX$SCAN_CSBS           ; Iterate over all CSBS
     3F 50        E9            BLBC    R0,DO_PHASE2            ; Branch when done
1F 60 A3    11    E1            BBC     #CSB$V_SELECTED, -      ; Branch if not selected
                                        CSB$L_STATUS(R3),10$
1A 60 A3    18    E0            BBS     #CSB$V_LOCAL, -         ; Branch if local CSB
                                        CSB$L_STATUS(R3),10$
15 60 A3    00    E0            BBS     #CSB$V_LONG_BREAK, -    ; Branch if permanent break
                                        CSB$L_STATUS(R3),10$    ;   in connection
        EF25'     30            BSBW    CNX$ALLOC_CDRP_ONLY
        EF22'     30            BSBW    CNX$RESOURCE_CHECK      ; Watch out for exhausted resources
     0D 50        E9            BLBC    R0,20$                  ; No memory available
                                ;
                                ; Send Phase 2 to one node
                                ;
50      0A        9A            MOVZBL  #CLMCNX$K_FNC_PH2,R0    ; Facility specific function code
      022B        30            BSBW    INIT_STD_MSG            ; Setup CDRP for sending message
      EF16'       30            BSBW    CNX$SEND_FORGET         ; Send message and forget it
      EF13'       31            BRW     CNX$SCAN_CSBS_FORK      ; Fork and resume scan

                  05            RSB                             ; Return and continue scan

      EF0F'       31            BRW     CNX$SCAN_CSBS_RETRY     ; Delay and resume scan
```

```
                     10F1  3633          .SBTTL  CNX$RCVD_PH2 - Phase 2 request received
                     10F1  3634
                     10F1  3635 ;++
                     10F1  3636 ;
                     10F1  3637 ; FUNCTIONAL DESCRIPTION:
                     10F1  3638 ;
                     10F1  3639 ;       This routine is called when a phase 2 request message is received.
                     10F1  3640 ;       The transition inprogress is committed.
                     10F1  3641 ;       No response or immediate acknowledgement is sent.
                     10F1  3642 ;
                     10F1  3643 ; CALLING SEQUENCE:
                     10F1  3644 ;
                     10F1  3645 ;       JSB     CNX$RCVD_PH2
                     10F1  3646 ;       IPL is IPL$_SCS
                     10F1  3647 ;
                     10F1  3648 ; INPUT PARAMETERS:
                     10F1  3649 ;
                     10F1  3650 ;       R2:     Message address
                     10F1  3651 ;       R3:     CSB of sending system
                     10F1  3652 ;       R4:     PDT address
                     10F1  3653 ;
                     10F1  3654 ; OUTPUT PARAMETERS:
                     10F1  3655 ;
                     10F1  3656 ;       NONE
                     10F1  3657 ;
                     10F1  3658 ; COMPLETION CODES:
                     10F1  3659 ;
                     10F1  3660 ;       NONE
                     10F1  3661 ;
                     10F1  3662 ; SIDE EFFECTS:
                     10F1  3663 ;
                     10F1  3664 ;       R0-R5 may be destroyed.
                     10F1  3665 ;--
                     10F1  3666
                     10F1  3667 CNX$RCVD_PH2::
   54    64 A3   D0  10F1  3668          MOVL    CSB$L_CLUB(R3),R4       ; Address of CLUB
07 1C A4    1D   E1  10F5  3669          BBC     #CLUB$V_TRANSITION, -   ; Branch if not locked and
                     10FA  3670                  CLUB$L_FLAGS(R4),10$    ;   bugcheck
48 A4    0C A2   D1  10FA  3671          CMPL    CLMCNX$L_XTN_ID(R2), -  ; Validate transaction ID
                     10FF  3672                  CLUB$L_COR_XTN(R4)
         04      13  10FF  3673          BEQL    20$                     ; Branch on match
                     1101  3674 10$:     BUG_CHECK       CNXMGRERR,FATAL ; Do something more sophisticated later?
                     1105  3675
         EEF8'   30  1105  3676 20$:     BSBW    CNX$DEALL_MSG_BUF_CSB   ; Deallocate message buffer
                     1108  3677
                     1108  3678 ;
                     1108  3679 ; Perform general transition completion operations
                     1108  3680 ;
                     1108  3681 DO_PHASE2:
34 A4    48 A4   D0  1108  3682          MOVL    CLUB$L_CUR_XTN(R4), -   ; Copy current transaction info into
                     110D  3683                  CLUB$L_LST_XTN(R4)
38 A4    4C A4   D0  110D  3684          MOVL    CLUB$L_CUR_COORD(R4), -
                     1112  3685                  CLUB$L_LST_COORD(R4)
3C A4    50 A4   7D  1112  3686          MOVQ    CLUB$Q_CUR_TIME(R4), -
                     1117  3687                  CLUB$Q_LST_TIME(R4)
44 A4    58 A4   90  1117  3688          MOVB    CLUB$B_CUR_CODE(R4), -
                     111C  3689                  CLUB$B_LST_CODE(R4)
```

```
                45 A4   59 A4   90  111C  3690          MOVB    CLUB$B_CUR_PHASE(R4), -
                                   1121  3691                  CLUB$B_LST_PHASE(R4)
                66 A4   64 A4   B0  1121  3692          MOVW    CLUB$W_NEXT_CSID(R4), - ; Update CSID allocation context
                                   1126  3693                  CLUB$W_FIRST_INDEX(R4)
                                   1126  3694 ;
                                   1126  3695 ; Case on transaction type and branch to appropriate place
                                   1126  3696 ;
                                   1126  3697          DISPATCH        CLUB$B_CUR_CODE(R4),TYPE=B,PREFIX=CLMCNX$K_XTN_, -
                                   1126  3698                  < -
                                   1126  3699                  <FORM,FORM_FINISH>, -   ; Cluster formation complete
                                   1126  3700                  <JOIN,JOIN_FINISH>, -   ; Joining cluster complete
                                   1126  3701                  <RECONFIG,RECONFIG_FINISH>, -   ; Failover complete
                                   1126  3702                  <QUORUM,QUORUM_FINISH>, - ; Quorum adjustment complete
                                   1126  3703                  >
                                   1133  3704          BUG_CHECK       CNXMGRERR,FATAL ; Invalid transaction code
                                   1137  3705
```

```
                     1137  3707          .SBTTL  LOCK_NODES - Obtain Coordinator Lock on Selected Nodes
                     1137  3708
                     1137  3709 ;++
                     1137  3710 ;
                     1137  3711 ; FUNCTIONAL DESCRIPTION:
                     1137  3712 ;
                     1137  3713 ;       This routine tries to obtain the coordinator lock on the
                     1137  3714 ;       selected nodes.
                     1137  3715 ;       The attempt fails if any node rejects the request or
                     1137  3716 ;       if the connection to a candidate node breaks.
                     1137  3717 ;       An immediate return is made to the caller's caller.
                     1137  3718 ;       If the locking operation is complete, a return is
                     1137  3719 ;       made to the caller.
                     1137  3720 ;       If the locking operation fails, the transition is aborted
                     1137  3721 ;       and no return is made; rather control passes to the unlock
                     1137  3722 ;       routine.
                     1137  3723 ;
                     1137  3724 ; CALLING SEQUENCE:
                     1137  3725 ;
                     1137  3726 ;       JSB     LOCK_NODES
                     1137  3727 ;       IPL is IPL$_SCS
                     1137  3728 ;
                     1137  3729 ; INPUT PARAMETERS:
                     1137  3730 ;
                     1137  3731 ;       CLUB$B_CUR_CODE gives transaction type
                     1137  3732 ;
                     1137  3733 ; OUTPUT PARAMETERS:
                     1137  3734 ;
                     1137  3735 ;       NONE
                     1137  3736 ;
                     1137  3737 ; COMPLETION CODES:
                     1137  3738 ;
                     1137  3739 ;       NONE
                     1137  3740 ;
                     1137  3741 ; SIDE EFFECTS:
                     1137  3742 ;
                     1137  3743 ;       If CSB is locked, the CSB$M_LOCKED bit in CSB$L_STATUS is set
                     1137  3744 ;       and the CSB$B_REF_CNT is incremented.
                     1137  3745 ;       R0-R5 are destroyed
                     1137  3746 ;
                     1137  3747 ;--
                     1137  3748
                     1137  3749 LOCK_NODES:
54   00000000'GF   D0   1137  3750          MOVL    G^CLU$GL_CLUB,R4            ; Address of CLUB
         74 A4 8ED0   113E  3751          POPL    CLUB$L_RET1(R4)             ; Save return address
      53   10 A4   D0   1142  3752          MOVL    CLUB$L_LOCAL_CSB(R4),R3     ; Address of local CSB
           EEB7'   30   1146  3753          BSBW    CNX$MARK_LOCKED             ; Mark local CSB locked first
           EEB4'   30   1149  3754          BSBW    CNX$CLUB_FORK               ; Release control briefly
         53   64   DE   114C  3755          MOVAL   CLUB$L_CSBQFL(R4),R3        ; Point to head of CSB list
         53   63   D0   114F  3756 10$:     MOVL    CSB$L_SYSQFL(R3),R3         ; Advance to next CSB
         54   53   D1   1152  3757          CMPL    R3,R4                       ; Done yet?
              3E   13   1155  3758          BEQL    90$                         ; Branch when done
   36 1C A4    11   E0   1157  3759 20$:     BBS     #CLUB$V_UNLOCK, -           ; Branch if unlock requested
                         115C  3760                  CLUB$L_FLAGS(R4),80$
   EE 60 A3    11   E1   115C  3761          BBC     #CSB$V_SELECTED, -          ; Ignore unselected nodes
                         1161  3762                  CSB$L_STATUS(R3),10$
   E9 60 A3    10   E0   1161  3763          BBS     #CSB$V_LOCKED, -            ; Branch if already locked
```

```
                        1166  3764                    CSB$L_STATUS(R3),10$
27 60 A3    00    E0    1166  3765            BBS     #CSB$V_LONG_BREAK, -    ; Branch if long break has been seen
                        116B  3766                    CSB$L_STATUS(R3),80$    ;   and unlock everything
          EE92'   30    116B  3767            BSBW    CNX$ALLOC_WARMCDRP_CSB
          EE8F'   30    116E  3768            BSBW    CNX$RESOURCE_CHECK      ; Watch out for exhausted resources
       05 50      E8    1171  3769            BLBS    R0,30$                  ; Branch if memory available
          EE89'   30    1174  3770            BSBW    CNX$CLUB_WAIT           ; Wait a second, saving R3
            DE    11    1177  3771            BRB     20$                     ; Restart scan
                        1179  3772
       50   03    9A    1179  3773 30$:       MOVZBL  #CLMCNX$K_FNC_LOCK,R0   ; Facility specific function code
          0193    30    117C  3774            BSBW    INIT_STD_MSG            ; Standard CDRP initialization
4C A5   98'AF    9E    117F  3775            MOVAB   B^BLD_LOCK_MSG, -       ; Message building routine
                        1184  3776                    CDRP$L_MSGBLD(R5)
          EE79'   30    1184  3777            BSBW    CNX$SEND_MSG_CSB        ; Send message
                        1187  3778 ;
                        1187  3779 ;       We are resumed here when the response message arrives.
                        1187  3780 ;       Registers contain:
                        1187  3781 ;               R0:     Status
                        1187  3782 ;               R2:     Address of message buffer
                        1187  3783 ;               R3:     Address of CSB
                        1187  3784 ;               R4:     Address of PDT
                        1187  3785 ;               R5:     Address of CDRP
                        1187  3786 ;
          015C    30    1187  3787            BSBW    CNX$PROCESS_RESPONSE    ; Deallocate storage, fetch status
       05 50      E9    118A  3788            BLBC    R0,80$                  ; Branch if ACK not received
                        118D  3789 ;
                        118D  3790 ; Setup data to lock node whose CSB is R3
                        118D  3791 ;
          EE70'   30    118D  3792            BSBW    CNX$MARK_LOCKED         ; Mark CSB locked
            BD    11    1190  3793            BRB     10$                     ; Continue with next CSB
                        1192  3794
          00B1    31    1192  3795 80$:       BRW     UNLOCK_ALL              ; Exit from transition
                        1195  3796
74 B4       17    1195  3797 90$:       JMP     @CLUB$L_RET1(R4)        ; Return to caller's caller
```

```
                        1198  3799          .SBTTL  BLD_LOCK_MSG - Build Coordinator Lock Request Message
                        1198  3800
                        1198  3801 ;++
                        1198  3802 ;
                        1198  3803 ; FUNCTIONAL DESCRIPTION:
                        1198  3804 ;
                        1198  3805 ;       This routine builds a message requesting the coordinator lock.
                        1198  3806 ;       The standard message building routine is used to build the message
                        1198  3807 ;       header.
                        1198  3808 ;
                        1198  3809 ; CALLING SEQUENCE:
                        1198  3810 ;
                        1198  3811 ;       JSB     BLD_LOCK_MSG
                        1198  3812 ;       IPL is IPL$_SCS
                        1198  3813 ;
                        1198  3814 ; INPUT PARAMETERS:
                        1198  3815 ;
                        1198  3816 ;       R2:     Address of message buffer
                        1198  3817 ;       R3:     Address of CSB
                        1198  3818 ;       R4:     Address of PDT
                        1198  3819 ;       R5:     Address of CDRP
                        1198  3820 ;               CDRP$L_VAL1(R5): Byte 0 contains facility code (negated for response
                        1198  3821 ;                                Byte 1 contains facility specific function code
                        1198  3822 ;               CDRP$L_VAL2(R5): Transition ID (from CLUB$L_CUR_XTN)
                        1198  3823 ;               CDRP$L_VAL3(R5): Byte 0 contains transition phase (from CLUB$B_CUR_P
                        1198  3824 ;                                Byte 1 contains transition code (from CLUB$B_XTN_CO
                        1198  3825 ;                                Byte 2 contains success/failure flag
                        1198  3826 ;                                Byte 3 contains reply code
                        1198  3827 ;               CDRP$L_VAL4(R5): Address of CSB to describe
                        1198  3828 ;
                        1198  3829 ; OUTPUT PARAMETERS:
                        1198  3830 ;
                        1198  3831 ;       NONE
                        1198  3832 ;
                        1198  3833 ; COMPLETION CODES:
                        1198  3834 ;
                        1198  3835 ;       NONE
                        1198  3836 ;
                        1198  3837 ; SIDE EFFECTS:
                        1198  3838 ;
                        1198  3839 ;       R0 and R1 are destroyed
                        1198  3840 ;
                        1198  3841 ;--
                        1198  3842
                        1198  3843 BLD_LOCK_MSG:
              019E      30  1198  3844          BSBW    BLD_STD_MSG             ; Fill in standard fields
         50   64 A3    D0  119B  3845          MOVL    CSB$L_CLUB(R3),R0       ; Address of CLUB
   14 A2      50 A0    7D  119F  3846          MOVQ    CLUB$Q_CUR_TIME(R0), -  ; Fill in transition time-stamp
                           11A4  3847                  CLMLCK$Q_XTN_TIME(R2)
                        05  11A4  3848          RSB
```

```
                        11A5  3850          .SBTTL  CNX$RCVD_LOCK - Lock request received
                        11A5  3851
                        11A5  3852 ;++
                        11A5  3853 ;
                        11A5  3854 ; FUNCTIONAL DESCRIPTION:
                        11A5  3855 ;
                        11A5  3856 ;       This routine is called when a lock request message is received.
                        11A5  3857 ;       A response must be sent which ACKs or NAKs the request.  The
                        11A5  3858 ;       request is NAKd if this node is already locked or in transition.
                        11A5  3859 ;
                        11A5  3860 ; CALLING SEQUENCE:
                        11A5  3861 ;
                        11A5  3862 ;       JSB     CNX$RCVD_LOCK
                        11A5  3863 ;       IPL is IPL$_SCS
                        11A5  3864 ;
                        11A5  3865 ; INPUT PARAMETERS:
                        11A5  3866 ;
                        11A5  3867 ;       R2:     Message address
                        11A5  3868 ;       R3:     CSB of sending system
                        11A5  3869 ;       R4:     PDT address
                        11A5  3870 ;       R5:     CDRP address (uninitialized)
                        11A5  3871 ;
                        11A5  3872 ; OUTPUT PARAMETERS:
                        11A5  3873 ;
                        11A5  3874 ;       NONE
                        11A5  3875 ;
                        11A5  3876 ; COMPLETION CODES:
                        11A5  3877 ;
                        11A5  3878 ;       NONE
                        11A5  3879 ;
                        11A5  3880 ; SIDE EFFECTS:
                        11A5  3881 ;
                        11A5  3882 ;       R0-R5 may be destroyed.
                        11A5  3883 ;--
                        11A5  3884
                        11A5  3885 CNX$RCVD_LOCK::
                   52   DD  11A5  3886          PUSHL   R2                              ; Save needed register
                EE56'   30  11A7  3887          BSBW    CNX$INIT_CDRP                   ; Initialize the CDRP for the response
          54    64 A3   D0  11AA  3888          MOVL    CSB$L_CLUB(R3),R4               ; Address of CLUB
                04      BA  11AE  3889          POPR    #^M<R2>                         ; Restore message address
50   68 A4    0C A2   C3  11B0  3890          SUBL3   CLMCNX$L_XTN_ID(R2), -          ; Update largest transaction seen -
                                                       CLUB$L_MAX_XTN(R4),R0           ;   status is only interesting result
                65      18  11B6  3892          BGEQ    10$                             ; No update needed
     68 A4    0C A2   D0  11B8  3893          MOVL    CLMCNX$L_XTN_ID(R2), -          ; Update local information
                                                       CLUB$L_MAX_XTN(R4)
     63 1C A4    13   E1  11BD  3895          BBC     #CLUB$V_INIT, -                 ; Branch if initialization not
                                                       CLUB$L_FLAGS(R4),20$            ;   complete and NAK request
     5E 1C A4    1D   E2  11C2  3897          BBSS    #CLUB$V_TRANSITION, -           ; Branch if busy and mark us busy
                                                       CLUB$L_FLAGS(R4), 20$
           0084 C4   D4  11C7  3899          CLRL    CLUB$L_TQE(R4)                  ; Abandon any timeout in progress
      50   00CC C4   9E  11CB  3900          MOVAB   CLUB$B_FORK_BLOCK(R4),R0        ; Address of transition fork block
     03 1C A0    00   E5  11D0  3901          BBCC    #CLUBFKB$V_FKB_BUSY, -          ; Branch if fork block is not
                                                       CLUBFKB$L_STATUS(R0),5$         ;   busy and mark no busy
           50    60   0F  11D5  3903          REMQUE  CLUBFKB$B_FORK_BLOCK(R0),R0     ; Remove from queue
                        11D8  3904 5$:
                        11D8  3905 ;
                        11D8  3906 ; Initialization for accepting lock request and beginning new transition
```

```
                                                11D8  3907 ;
                                    EE25'  30   11D8  3908          BSBW    CNX$INIT_CSBS                   ; Initialize all CSBs for new transition
          1C A4     00020000 8F     CA   11DB  3909          BICL2   #CLUB$M_ONLOCK, -                ; Clear unlock flag
                                                11E3  3910                  CLUB$L_FLAGS(R4)
                       50   10 A4   D0   11E3  3911          MOVL    CLUB$L_LOCAL_CSB(R4),R0 ; Address of local CSB
          60 A0     00020000 8F     C8   11E7  3912          BISL2   #CSB$M_SELECTED, -              ; Mark local node SELECTED
                                                11EF  3913                  CSB$L_STATUS(R0)
          60 A3     00020000 8F     C8   11EF  3914          BISL2   #CSB$M_SELECTED, -              ; Mark coordinator SELECTED
                                                11F7  3915                  CSB$L_STATUS(R3)
                 48 A4    0C A2     D0   11F7  3916          MOVL    CLMCNX$L_XTN_ID(R2), -          ; Update local information about
                                                11FC  3917                  CLUB$L_COR_XTN(R4)      ;   transition
                 4C A4    4C A3     D0   11FC  3918          MOVL    CSB$L_CSID(R3), -               ; CSID of coordinator
                                                1201  3919                  CLUB$L_CUR_COORD(R4)
                 58 A4    11 A2     90   1201  3920          MOVB    CLMCNX$B_XTN_CODE(R2), - ; Transition type
                                                1206  3921                  CLUB$B_COR_CODE(R4)
                 59 A4    10 A2     90   1206  3922          MOVB    CLMCNX$B_XTN_PHASE(R2), - ; Transition phase
                                                120B  3923                  CLUB$B_COR_PHASE(R4)
                 50 A4    14 A2     7D   120B  3924          MOVQ    CLMLCK$Q_XTN_TIME(R2), - ; Transition time-stamo
                                                1210  3925                  CLUB$Q_COR_TIME(R4)
                    5C A4    53     D0   1210  3926          MOVL    R3,CLUB$L_COORD(R4)             ; CSB of coordinator
                 64 A4    66 A4     B0   1214  3927          MOVW    CLUB$W_FIRST_INDEX(R4), - ; Initialize CSID allocation context
                                                1219  3928                  CLUB$W_NEXT_CSID(R4)
                                    01   DD   1219  3929          PUSHL   #1                              ; Accept request
                                    0A   11   121B  3930          BRB     30$
                                                121D  3931 ;
                                                121D  3932 ; Transition number is invalid.
                                                121D  3933 ; Lock request will be rejected.
                                                121D  3934 ; Request that status be send to remote node.
                                                121D  3935 ;
          60 A3     04000000 8F     C8   121D  3936 10$:    BISL2   #CSB$M_SEND_STATUS, -   ; Send status request
                                                1225  3937                  CSB$L_STATUS(R3)
                                                1225  3938 ;
                                                1225  3939 ; Common reject point
                                                1225  3940 ;
                       00   DD       1225  3941 20$:    PUSHL   #0                              ; Reject lock request
                    50   03   9A     1227  3942 30$:    MOVZBL  #CLMCNX$K_FNC_LOCK,R0   ; Facility specific function code
                       00DC 30       122A  3943          BSBW    CNX$INIT_STD_RESP               ; Init CDRP for standard response
                36 A5   8E   F6     122D  3944          CVTLB   (SP)+,CDRP$L_VAL3+2(R5) ; Store success/failure flag
                       EDCC' 31     1231  3945          BRW     CNX$RESP_FORGET                 ; Send response and forget it
```

```
                                                E 4
                        1234  3947          .SBTTL  CNX$CHECK_UNLOCK - Check UNLOCK flag
                        1234  3948
                        1234  3949 ;++
                        1234  3950 ;
                        1234  3951 ; FUNCTIONAL DESCRIPTION:
                        1234  3952 ;
                        1234  3953 ;       Test the UNLOCK flag in the CLUB.  If it is clear, return to the caller.
                        1234  3954 ;       If it is set, remove the caller's PC from the stack, and branch to UNLOCK_AL
                        1234  3955 ;       This will eventually return to the caller's caller.
                        1234  3956 ;
                        1234  3957 ; CALLING SEQUENCE:
                        1234  3958 ;
                        1234  3959 ;       JSB     CNX$CHECK_UNLOCK
                        1234  3960 ;       IPL is IPL$_SCS
                        1234  3961 ;
                        1234  3962 ; INPUT PARAMETERS:
                        1234  3963 ;
                        1234  3964 ;       The CLUB$M_UNLOCK bit CLUB$L_FLAGS is the primary input
                        1234  3965 ;
                        1234  3966 ; OUTPUT PARAMETERS:
                        1234  3967 ;
                        1234  3968 ;       R4 contains the address of the CLUB
                        1234  3969 ;
                        1234  3970 ; COMPLETION CODES:
                        1234  3971 ;
                        1234  3972 ;       NONE
                        1234  3973 ;
                        1234  3974 ; SIDE EFFECTS:
                        1234  3975 ;
                        1234  3976 ;       NONE
                        1234  3977 ;
                        1234  3978 ;--
                        1234  3979
                        1234  3980 CNX$CHECK_UNLOCK::
54   00000000'GF   D0   1234  3981          MOVL    G^CLU$GL_CLUB,R4        ; Get address of CLUB
   01 1C A4    11   E0   123B  3982          BBS     #CLUB$V_UNLOCK, -       ; Branch if unlock is request and abort
                        1240  3983                  CLUB$L_FLAGS(R4),10$    ;   this transition
                   05   1240  3984          RSB
                        1241  3985
        5E    04   C0   1241  3986 10$:     ADDL2   #4,SP                   ; Remove caller's PC
              00   11   1244  3987          BRB     UNLOCK_ALL              ; Abort transition and return to caller's ca
```

```
                                                F 4
                     1246  3989          .SBTTL  UNLOCK_ALL - Release Coordinator Lock on All Nodes
                     1246  3990
                     1246  3991 ;++
                     1246  3992 ;
                     1246  3993 ; FUNCTIONAL DESCRIPTION:
                     1246  3994 ;
                     1246  3995 ;       This routine releases the coordinator lock on all nodes.
                     1246  3996 ;       A breaking link counts as a successful release of the lock.
                     1246  3997 ;
                     1246  3998 ; CALLING SEQUENCE:
                     1246  3999 ;
                     1246  4000 ;       JSB     UNLOCK_ALL
                     1246  4001 ;       IPL is IPL$_SCS
                     1246  4002 ;
                     1246  4003 ; INPUT PARAMETERS:
                     1246  4004 ;
                     1246  4005 ;       CLUB$B_CUR_CODE gives transaction type
                     1246  4006 ;
                     1246  4007 ; OUTPUT PARAMETERS:
                     1246  4008 ;
                     1246  4009 ;       NONE
                     1246  4010 ;
                     1246  4011 ; COMPLETION CODES:
                     1246  4012 ;
                     1246  4013 ;       NONE
                     1246  4014 ;
                     1246  4015 ; SIDE EFFECTS:
                     1246  4016 ;
                     1246  4017 ;       R0-R5 are destroyed
                     1246  4018 ;
                     1246  4019 ;--
                     1246  4020
                     1246  4021 UNLOCK_ALL:
            EDB7'    30  1246  4022          BSBW    CNX$CHECK_QUORUM        ; Adjust dynamic quorum flag, hang if no quo
54   00000000'GF     D0  1249  4023          MOVL    G^CLU$GL_CLUB,R4        ; Address of CLUB
     53    10 A4     D0  1250  4024          MOVL    CLUB$L_LOCAL_CSB(R4),R3 ; Local CSB address
  10 60 A3    10     E0  1254  4025          BBS     #CSB$V_LOCKED, -        ; Branch if still locked
                         1259  4026                  CSB$L_STATUS(R3),5$
     5C A4    53     D1  1259  4027          CMPL    R3,CLUB$L_COORD(R4)     ; Is local system the coordinator?
              0A     12  125D  4028          BNEQ    5$                      ; Branch if not coordinator
              55     D4  125F  4029          CLRL    R5                      ; No CSB address for message
   50    0000'CF     9E  1261  4030          MOVAB   COMPLETE_MSG,R0         ; Cluster state transition complete message
            ED97'    30  1266  4031          BSBW    CNX$CONFIG_CHANGE       ; Make this state known
                         1269  4032 5$:
  59 A4    50 8F     90  1269  4033          MOVB    #CLMCNX$K_UNLOCK, -     ; Set phase to UNLOCK
                         126E  4034                  CLUB$B_CUR_PHASE(R4)
            ED8F'    30  126E  4035          BSBW    CNX$SCAN_CSBS           ; Iterate over all CSB's
          4E 50      E9  1271  4036          BLBC    R0,UNLOCK_NODE          ; Branch when done
  05 60 A3    01     E0  1274  4037          BBS     #CSB$V_MEMBER, -        ; Branch if a cluster member
                         1279  4038                  CSB$L_STATUS(R3),10$
  00 60 A3    1A     E3  1279  4039          BBCS    #CSB$V_SEND_STATUS, -   ; Request status for non-cluster nodes
                         127E  4040                  CSB$L_STATUS(R3), 10$   ;   informing them of this nodes' new status
  35 60 A3    10     E1  127E  4041 10$:     BBC     #CSB$V_LOCKED, -        ; Branch if not locked
                         1283  4042                  CSB$L_STATUS(R3),40$
  5C A4    10 A4     D1  1283  4043          CMPL    CLUB$L_LOCAL_CSB(R4), - ; Is local system the
                         1288  4044                  CLUB$L_COORD(R4)        ;   coordinator?
              28     12  1288  4045          BNEQ    30$                     ; Branch if not coordinator
```

```
                19 60 A3   18   E0  128A  4046          BBS     #CSB$V_LOCAL, -          ; Branch if local CSB
                                   128F  4047                  CSB$L_STATUS(R3),20$
                1E 60 A3   00   E0  128F  4048          BBS     #CSB$V_LONG_BREAK, -     ; Branch if connection really
                                   1294  4049                  CSB$L_STATUS(R3),30$     ;   broken
                       ED69'   30   1294  4050          BSBW    CNX$ALLOC_CDRP_ONLY      ; Get a CDRP
                       ED66'   30   1297  4051          BSBW    CNX$RESOURCE_CHECK       ; Watch out for exhausted resources
                     1C 50     E9   129A  4052          BLBC    R0,50$                   ; No memory available
                   50    04   9A   129D  4053          MOVZBL  #CLMCNX$K_FNC_UNLOCK,R0  ; Facility specific function code
                      006F   30   12A0  4054          BSBW    INIT_STD_MSG             ; Init CDRP for standard message
                      ED5A'   30   12A3  4055          BSBW    CNX$SEND_FORGET          ; Send message and forget it
                        0A    11   12A6  4056          BRB     30$                      ; Branch to common exit
                                   12A8  4057
                                   12A8  4058 ; Output ABORT message only once -- when processing local node
                                   12A8  4059 ;
                        55    D4   12A8  4060 20$:     CLRL    R5                       ; No CSB address
                 50   0000'CF    9E   12AA  4061          MOVAB   UNLOCK_MSG,R0            ; Aborting transition message
                      ED4E'   30   12AF  4062          BSBW    CNX$CONFIG_CHANGE        ; Make this state known
                                   12B2  4063
                                   12B2  4064 ; Common exit
                                   12B2  4065 ;
                      ED4B'   30   12B2  4066 30$:     BSBW    CNX$MARK_UNLOCKED        ; Mark CSB not locked.  If the CSB
                                   12B5  4067                                           ;   is deleted, R3 is previous list entry
                      ED48'   31   12B5  4068          BRW     CNX$SCAN_CSBS_FORK       ; Fork and then continue scan
                                   12B8  4069
                        05   12B8  4070 40$:     RSB
                                   12B9  4071
                      ED44'   31   12B9  4072 50$:     BRW     CNX$SCAN_CSBS_RETRY      ; Delay and then resume scan
```

```
                        12BC  4074          .SBTTL  CNX$RCVD_UNLOCK - Unlock request received
                        12BC  4075
                        12BC  4076 ;++
                        12BC  4077 ;
                        12BC  4078 ; FUNCTIONAL DESCRIPTION:
                        12BC  4079 ;
                        12BC  4080 ;       This routine is called when an unlock request message is received.
                        12BC  4081 ;       The node is unlocked. No response or immediate acknowledgement is sent.
                        12BC  4082 ;
                        12BC  4083 ; CALLING SEQUENCE:
                        12BC  4084 ;
                        12BC  4085 ;       JSB     CNX$RCVD_UNLOCK
                        12BC  4086 ;       IPL is IPL$_SCS
                        12BC  4087 ;
                        12BC  4088 ; INPUT PARAMETERS:
                        12BC  4089 ;
                        12BC  4090 ;       R2:     Message address
                        12BC  4091 ;       R3:     CSB of sending system
                        12BC  4092 ;       R4:     PDT address
                        12BC  4093 ;
                        12BC  4094 ; OUTPUT PARAMETERS:
                        12BC  4095 ;
                        12BC  4096 ;       NONE
                        12BC  4097 ;
                        12BC  4098 ; COMPLETION CODES:
                        12BC  4099 ;
                        12BC  4100 ;       NONE
                        12BC  4101 ;
                        12BC  4102 ; SIDE EFFECTS:
                        12BC  4103 ;
                        12BC  4104 ;       R0-R5 may be destroyed.
                        12BC  4105 ;--
                        12BC  4106
                        12BC  4107 CNX$RCVD_UNLOCK::
        008A    30      12BC  4108          BSBW    MSG_CHECK               ; Validate message
        ED3E'   30      12BF  4109          BSBW    CNX$DEALL_MSG_BUF_CSB   ; Deallocate message buffer
                        12C2  4110 ;
                        12C2  4111 ; Right here, all nodes have been unlocked.
                        12C2  4112 ; May need to start up some activity here.
                        12C2  4113 ;
                        12C2  4114 UNLOCK_NODE:
20 A4   22 A4   B1      12C2  4115          CMPW    CLUB$W_VOTES(R4), -     ; Is there a quorum?
                        12C7  4116                  CLUB$W_QUORUM(R4)
           09   1E      12C7  4117          BGEQU   10$                     ; Branch if yes
04 1C A4   02   E1      12C9  4118          BBC     #CLUB$V_SHUTDOWN, -     ; Branch if not in shutdown mode
                        12CE  4119                  CLUB$L_FLAGS(R4),10$
                        12CE  4120          BUG_CHECK       OPERATOR,FATAL  ; Finish cluster-wide shutdown
                        12D2  4121
00 1C A4   1B   E5      12D2  4122 10$:     BBCC    #CLUB$V_ADJ_QUORUM, -   ; Clear quorum adjustment flag
                        12D7  4123                  CLUB$L_FLAGS(R4),20$
   59 A4   10   90      12D7  4124 20$:     MOVB    #CLMCNX$K_IDLE, -       ; Set phase to IDLE
                        12DB  4125                  CLUB$B_CUR_PHASE(R4)
00 1C A4   1D   E5      12DB  4126          BBCC    #CLUB$V_TRANSITION, -   ; Clear transition flag
                        12E0  4127                  CLUB$L_FLAGS(R4),30$
        ED1D'   30      12E0  4128 30$:     BSBW    CNX$CHECK_QUORUM        ; Check dynamic quorum
        F1D0    31      12E3  4129          BRW     START_TIMEOUT           ; Keep the home fires burning...
```

```
                        12E6  4131          .SBTTL  CNX$PROCESS_RESPONSE - Process simple response message
                        12E6  4132
                        12E6  4133 ;++
                        12E6  4134 ;
                        12E6  4135 ; FUNCTIONAL DESCRIPTION:
                        12E6  4136 ;
                        12E6  4137 ;       Process a reponse message by deallocating the message buffer
                        12E6  4138 ;       and CDRP.
                        12E6  4139 ;       Return status indicating whether the connection broke, the
                        12E6  4140 ;       response was an ACK, or the response was an ACK.
                        12E6  4141 ;
                        12E6  4142 ; CALLING SEQUENCE:
                        12E6  4143 ;
                        12E6  4144 ;       JSB     CNX$PROCESS_RESPONSE
                        12E6  4145 ;       IPL is IPL$_SCS
                        12E6  4146 ;
                        12E6  4147 ; INPUT PARAMETERS:
                        12E6  4148 ;
                        12E6  4149 ;       R0:     Status
                        12E6  4150 ;       R2:     Address of message buffer
                        12E6  4151 ;       R3:     Address of CSB
                        12E6  4152 ;       R4:     Address of PDT
                        12E6  4153 ;       R5:     Address of CDRP
                        12E6  4154 ;
                        12E6  4155 ; OUTPUT PARAMETERS:
                        12E6  4156 ;
                        12E6  4157 ;       R4:     Address of CLUB
                        12E6  4158 ;
                        12E6  4159 ; COMPLETION CODES:
                        12E6  4160 ;
                        12E6  4161 ;       R0:
                        12E6  4162 ;               1 (success) Response was ACK
                        12E6  4163 ;               0 (failure) Response was NAK
                        12E6  4164 ;               2 (failure) Connection broke
                        12E6  4165 ;
                        12E6  4166 ; SIDE EFFECTS:
                        12E6  4167 ;
                        12E6  4168 ;       R1, R2, and R5 are destroyed
                        12E6  4169 ;
                        12E6  4170 ;--
                        12E6  4171
                        12E6  4172 CNX$PROCESS_RESPONSE::
      08 50    E9       12E6  4173          BLBC    R0,10$                  ; Error sending message
7E    12 A2    9A       12E9  4174          MOVZBL  CLM$CNX$B_ACK(R2),-(SP) ; Save response
       ED10'   30       12ED  4175          BSBW    CNX$DEALC_WARMCDRP_CSB  ; Recycle warm CDRP
         01    BA       12F0  4176          POPR    #^M<R0>                 ; Restore response ACK/NAK
         10    11       12F2  4177          BRB     20$                     ; Branch to common exit
                        12F4  4178 ;
         53    DD       12F4  4179 10$:     PUSHL   R3                      ; Save CSB address
   50    55    D0       12F6  4180          MOVL    R5,R0                   ; Address of CDRP
00000000'GF    16       12F9  4181          JSB     G^EXE$DEANONPAGED       ; Deallocate CDRP
         08    BA       12FF  4182          POPR    #^M<R3>                 ; Restore CSB address
   50    02    D0       1301  4183          MOVL    #2,R0                   ; Return status code
54    64 A3    D0       1304  4184 20$:     MOVL    CSB$L_CLUB(R3),R4       ; Address of CLUB
               05       1308  4185          RSB                             ; Return to caller
```

```
                                J 4
                     1309  4187          .SBTTL  INIT_STD_MSG - Common CDRP initialization for standard message
                     1309  4188          .SBTTL  CNX$INIT_STD_RESP - Common CDRP initialization for standard response
                     1309  4189
                     1309  4190 ;++
                     1309  4191 ;
                     1309  4192 ; FUNCTIONAL DESCRIPTION:
                     1309  4193 ;
                     1309  4194 ;       This routine initializes most of the CDRP fields used by the standard
                     1309  4195 ;       message build routines.  Two entry points are provided -- one for
                     1309  4196 ;       normal messages and one for responses.
                     1309  4197 ;       These routines are provided for use in conjunction with the BLD_STD_MSG
                     1309  4198 ;       message building routine.
                     1309  4199 ;
                     1309  4200 ; CALLING SEQUENCE:
                     1309  4201 ;
                     1309  4202 ;       JSB     INIT_STD_MSG
                     1309  4203 ;       JSB     CNX$INIT_STD_RESP
                     1309  4204 ;       IPL is IPL$_SCS
                     1309  4205 ;
                     1309  4206 ; INPUT PARAMETERS:
                     1309  4207 ;
                     1309  4208 ;       R0:     Message specified data
                     1309  4209 ;               Byte 0: Facility specific function code
                     1309  4210 ;               Byte 2: ACK byte data (INIT_STD_MSG only)
                     1309  4211 ;               Byte 3: REPLY byte data (INIT_STD_MSG only)
                     1309  4212 ;       R3:     Address of CSB
                     1309  4213 ;       R4:     Address of CLUB
                     1309  4214 ;               CLUB$L_CUR_XTN(R4): Current transition id
                     1309  4215 ;               CLUB$B_CUR_PHASE(R4): Current transition phase
                     1309  4216 ;               CLUB$B_CUR_CODE(R4): Current transition code
                     1309  4217 ;       R5:     Address of CDRP
                     1309  4218 ;
                     1309  4219 ; OUTPUT PARAMETERS:
                     1309  4220 ;
                     1309  4221 ;       NONE
                     1309  4222 ;
                     1309  4223 ; COMPLETION CODES:
                     1309  4224 ;
                     1309  4225 ;       NONE
                     1309  4226 ;
                     1309  4227 ; SIDE EFFECTS:
                     1309  4228 ;
                     1309  4229 ;       R0 and R1 are destroyed
                     1309  4230 ;
                     1309  4231 ;--
                     1309  4232
                     1309  4233          .ENABLE LSB
                     1309  4234
                     1309  4235 CNX$INIT_STD_RESP::
51   81 8F   90      1309  4236          MOVB    #<CLSMSG$K_FAC_CNX ! -  ; Facility message code
                     130D  4237                    CLSMSG$M_RESPMSG>,R1  ;   with response tag
   50   50   9A      130D  4238          MOVZBL  R0,R0                   ; Ignore high order bits *** temp
        03   11      1310  4239          BRB     10$
                     1312  4240
                     1312  4241 INIT_STD_MSG:
   51   01   90      1312  4242          MOVB    #CLSMSG$K_FAC_CNX,R1    ; Facility message code
     2C A5   D4      1315  4243 10$:     CLRL    CDRP$L_VA[1(R5)         ; Initialize other fields
```

```
                2C A5    51    90  1318  4244          MOVB    R1,CDRP$L_VAL1+0(R5)    ; Facility identification code
                2D A5    50    90  131C  4245          MOVB    R0,CDRP$L_VAL1+1(R5)    ; Facility specific code
            30 A5   48 A4      D0  1320  4246          MOVL    CLUB$L_CUR_XTN(R4), -   ; Current transition ID
                                   1325  4247                  CDRP$L_VAL2(R5)
                34 A5    50    D0  1325  4248          MOVL    R0,CDRP$L_VAL3(R5)      ; Store byte 2 of R0 as ack  field
                                   1329  4249                                          ;   and byte 3 of R0 as reply field
            34 A5   59 A4      90  1329  4250          MOVB    CLUB$B_CUR_PHASE(R4), - ; Current transition phase
                                   132E  4251                  CDRP$L_VAL3(R5)
            35 A5   58 A4      90  132E  4252          MOVB    CLUB$B_CUR_CODE(R4), -  ; Current transition code
                                   1333  4253                  CDRP$L_VAL3+1(R5)
            4C A5   39'AF      9E  1333  4254          MOVAB   B^BLD_STD_MSG, -        ; Standard message build routine
                                   1338  4255                  CDRP$C_MSGBLD(R5)
                                   1338  4256          RSB
                                   1339  4257
                                   1339  4258          .DISABLE        LSB
```

```
                     1339  4260          .SBTTL  BLD_STD_MSG - Build Standard Message from CDRP
                     1339  4261
                     1339  4262 ;++
                     1339  4263 ;
                     1339  4264 ; FUNCTIONAL DESCRIPTION:
                     1339  4265 ;
                     1339  4266 ;       This routine is a common message build routine that fills in
                     1339  4267 ;       the common message header from data in the CDRP.
                     1339  4268 ;       There is no dependence upon the contents of the CLUB in building
                     1339  4269 ;       the message.  This means that this routine is safe to use for
                     1339  4270 ;       messages that are queued and forgotten and which can be sent
                     1339  4271 ;       at a later time when the state of the world has changed.
                     1339  4272 ;
                     1339  4273 ; CALLING SEQUENCE:
                     1339  4274 ;
                     1339  4275 ;       JSB     BLD_STD_MSG
                     1339  4276 ;       IPL is IPL$_SCS
                     1339  4277 ;
                     1339  4278 ; INPUT PARAMETERS:
                     1339  4279 ;
                     1339  4280 ;       R2:     Address of message buffer
                     1339  4281 ;       R3:     Address of CSB
                     1339  4282 ;       R4:     Address of PDT
                     1339  4283 ;       R5:     Address of CDRP
                     1339  4284 ;               CDRP$L_VAL1(R5): Byte 0 contains facility code (negated for response
                     1339  4285 ;                                Byte 1 contains facility specific function code
                     1339  4286 ;               CDRP$L_VAL2(R5): Transition ID (from CLUB$L_CUR_XTN)
                     1339  4287 ;               CDRP$L_VAL3(R5): Byte 0 contains transition phase (from CLUB$B_CUR_P
                     1339  4288 ;                                Byte 1 contains transition code (from CLUB$B_XTN_CO
                     1339  4289 ;                                Byte 2 contains success/failure flag
                     1339  4290 ;                                Byte 3 contains reply code
                     1339  4291 ;
                     1339  4292 ; OUTPUT PARAMETERS:
                     1339  4293 ;
                     1339  4294 ;       NONE
                     1339  4295 ;
                     1339  4296 ; COMPLETION CODES:
                     1339  4297 ;
                     1339  4298 ;       NONE
                     1339  4299 ;
                     1339  4300 ; SIDE EFFECTS:
                     1339  4301 ;
                     1339  4302 ;       R0 and R1 are destroyed
                     1339  4303 ;
                     1339  4304 ;--
                     1339  4305
                     1339  4306 BLD_STD_MSG:
08 A2    2C A5   90  1339  4307          MOVB    CDRP$L_VAL1+0(R5), -    ; Facility message / response code
                     133E  4308                  CLSMSG$B_FACILITY(R2)
09 A2    2D A5   90  133E  4309          MOVB    CDRP$L_VAL1+1(R5), -    ; Facility specific function code
                     1343  4310                  CLSMSG$B_FUNC(R2)
                     1343  4311          ASSUME  CLMCNX$B_XTN_PHASE EQ CLMCNX$L_XTN_ID+4
                     1343  4312          ASSUME  CDRP$L_VAL3 EQ CDRP$L_VAL2+4
                     1343  4313          ASSUME  CLMCNX$B_XTN_CODE EQ CLMCNX$B_XTN_PHASE+1
                     1343  4314          ASSUME  CLMCNX$B_ACK EQ CLMCNX$B_XTN_CODE+1
                     1343  4315          ASSUME  CLMCNX$B_REPLY EQ CLMCNX$B_ACK+1
0C A2    30 A5   7D  1343  4316          MOVQ    CDRP$L_VAL2(R5), -      ; Standard information
```

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

```
                     1349  4320          .SBTTL  MSG_CHECK - Standard incoming message verification
                     1349  4321
                     1349  4322 ;++
                     1349  4323 ;
                     1349  4324 ; FUNCTIONAL DESCRIPTION:
                     1349  4325 ;
                     1349  4326 ;       This routine is called to perform standard checks upon an
                     1349  4327 ;       incoming message.  These checks may not be application to
                     1349  4328 ;       messages intitiating or completing a transition.
                     1349  4329 ;
                     1349  4330 ; CALLING SEQUENCE:
                     1349  4331 ;
                     1349  4332 ;       JSB     MSG_CHECK
                     1349  4333 ;       IPL is IPL$_SCS
                     1349  4334 ;
                     1349  4335 ; INPUT PARAMETERS:
                     1349  4336 ;
                     1349  4337 ;       R2:     Message address
                     1349  4338 ;       R3:     CSB of sending system
                     1349  4339 ;       R4:     PDT address
                     1349  4340 ;       R5:     CDRP address (uninitialized)
                     1349  4341 ;
                     1349  4342 ; OUTPUT PARAMETERS:
                     1349  4343 ;
                     1349  4344 ;       R4:     Address of the CLUB
                     1349  4345 ;
                     1349  4346 ; COMPLETION CODES:
                     1349  4347 ;
                     1349  4348 ;       NONE
                     1349  4349 ;
                     1349  4350 ; SIDE EFFECTS:
                     1349  4351 ;
                     1349  4352 ;       R0-R1 may be destroyed.
                     1349  4353 ;--
                     1349  4354
                     1349  4355 MSG_CHECK:
    54   64 A3   D0  1349  4356          MOVL    CSB$L_CLUB(R3),R4       ; Address of CLUB
48 A4    0C A2   D1  134D  4357          CMPL    CLMCNX$L_XTN_ID(R2), -  ; Check for inconsistency
                     1352  4358                  CLUB$L_COR_XTN(R4)
              0B 12  1352  4359          BNEQ    10$                     ; Bugcheck on inconsistency
06 1C A4      1D E1  1354  4360          BBC     #CLUB$V_TRANSITION, -   ; Branch on inconsistency
                     1359  4361                  CLUB$L_FLAGS(R4), 10$   ;   and bugcheck
59 A4    10 A2   90  1359  4362          MOVB    CLMCNX$B_XTN_PHASE(R2), - ; Save transition phase
                     135E  4363                  CLUB$B_COR_PHASE(R4)
                 05  135E  4364          RSB
                     135F  4365
                     135F  4366 10$:     BUG_CHECK       CNXMGRERR,FATAL ; Inconsistent message
```

```
                        1363  4368          .SBTTL  UPDATE_QUORUM - Update Quorum Parameters in CLUB
                        1363  4369
                        1363  4370 ;++
                        1363  4371 ;
                        1363  4372 ; FUNCTIONAL DESCRIPTION:
                        1363  4373 ;
                        1363  4374 ;       Update the CLUB$W_QUORUM, CLUB$W_VOTES, CLUB$W_QDVOTES, and
                        1363  4375 ;       CLUB$W_NODES fields of the CLUB at the conclusion of a fully
                        1363  4376 ;       synchronized state change.
                        1363  4377 ;
                        1365  4378 ; CALLING SEQUENCE:
                        1363  4379 ;
                        1363  4380 ;       JSB     UPDATE_QUORUM
                        1363  4381 ;       IPL is IPL$_SCS
                        1363  4382 ;
                        1363  4383 ; INPUT PARAMETERS:
                        1363  4384 ;
                        1363  4385 ;       CSB'S in list with CSB$M_SELECTED bit set
                        1363  4386 ;       CLUB$W_NEWQUORUM is quorum
                        1363  4387 ;       CLUB$W_NEWQDVOTES is votes assigned to quorum disk
                        1363  4388 ;       CLUB$V_QF_NEWVOTE is quorum disk membership flag
                        1363  4389 ;
                        1363  4390 ; OUTPUT PARAMETERS:
                        1363  4391 ;
                        1363  4392 ;       R4 is address of CLUB
                        1363  4393 ;       CLUB$W_QUORUM is new quorum
                        1363  4394 ;       CLUB$W_VOTES is new number of votres present in the cluster
                        1363  4395 ;       CLUB$W_NODES is number of nodes in cluster
                        1363  4396 ;       CLUB$V_QF_VOTE indicates that the quorum disk is a member
                        1363  4397 ;
                        1363  4398 ; COMPLETION CODES:
                        1363  4399 ;
                        1363  4400 ;       NONE
                        1363  4401 ;
                        1363  4402 ; SIDE EFFECTS:
                        1363  4403 ;
                        1363  4404 ;       R0 and R1 are destroyed.
                        1363  4405 ;--
                        1363  4406
                        1363  4407 UPDATE_QUORUM:
                    0C  BB  1363  4408          PUSHR   #^M<R2,R3>              ; Save registers
54  00000000'GF    D0  1365  4409          MOVL    G^CLU$GL_CLUB,R4        ; Address of CLUB
 20 A4   00A4 C4    B0  136C  4410          MOVW    CLUB$W_NEWQUORUM(R4), - ; Update cluster quorum
                        1372  4411                  CLUB$W_QUORUM(R4)
 00AE C4   46 A4    B0  1372  4412          MOVW    CLUB$W_NEWQDVOTES(R4), - ; Update quorum disk votes
                        1378  4413                  CLUB$W_QDVOTES(R4)
           24 A4    B4  1378  4414          CLRW    CLUB$W_NODES(R4)        ; Initialize count of nodes
           22 A4    B4  137B  4415          CLRW    CLUB$W_VOTES(R4)        ; Initialize vote count
1C A4   02000000 8F    CA  137E  4416          BICL    #CLUB$M_QF_VOTE, -      ; Assume no quorum disk membership
                        1386  4417                  CLUB$L_FLAGS(R4)
    0A 1C A4    1A    E1  1386  4418          BBC     #CLUB$V_QF_NEWVOTE, -   ; Branch if no quorum disk membership
                        138B  4419                  CLUB$L_FLAGS(R4),10$
    22 A4   46 A4    A0  138B  4420          ADDW    CLUB$W_NEWQDVOTES(R4), - ; Count quorum disk in votes
                        1390  4421                  CLUB$W_VOTES(R4)
    00 1C A4    19    E2  1390  4422          BBSS    #CLUB$V_QF_VOTE, -      ; Mark quorum disk a member
                        1395  4423                  CLUB$L_FLAGS(R4),10$
           EC68'    30  1395  4424 10$:     BSBW    CNX$SCAN_CSBS           ; Iterate over all CSBs
```

```
                OE 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  4436          .SBTTL  ADD_NODE - Make node a cluster member
                                13AC  4437 ;++
                                13AC  4438 ;
                                13AC  4439 ; FUNCTIONAL DESCRIPTION:
                                13AC  4440 ;
                                13AC  4441 ;       Make the specified node a cluster member.
                                13AC  4442 ;
                                13AC  4443 ; CALLING SEQUENCE:
                                13AC  4444 ;
                                13AC  4445 ;       JSB     ADD_NODE
                                13AC  4446 ;       IPL is IPL$_SCS
                                13AC  4447 ;
                                13AC  4448 ; INPUT PARAMETERS:
                                13AC  4449 ;
                                13AC  4450 ;       R3:     CSB of the node to be added
                                13AC  4451 ;               CSB$L_CSID contains CSID for the node
                                13AC  4452 ;
                                13AC  4453 ; OUTPUT PARAMETERS:
                                13AC  4454 ;
                                13AC  4455 ;       R4:     Address of CLUB
                                13AC  4456 ;
                                13AC  4457 ; COMPLETION CODES:
                                13AC  4458 ;
                                13AC  4459 ;       NONE
                                13AC  4460 ;
                                13AC  4461 ; SIDE EFFECTS:
                                13AC  4462 ;
                                13AC  4463 ;       The cluster vector is updated.
                                13AC  4464 ;       The CSB reference count is incremented.
                                13AC  4465 ;       R0-R1 are destroyed.
                                13AC  4466 ;
                                13AC  4467 ;--
                                13AC  4468
                                13AC  4469 ADD_NODE:
          6C A3    96           13AC  4470          INCB    CSB$B_REF_CNT(R3)       ; Bump reference count to nail down CSB
       60 A3    02    C8        13AF  4471          BISL?   #CSB$M_MEMBER, -        ; Mark the node a cluster member
                                13B3  4472                  CSB$L_STATUS(R3)
       50    4C A3    3C        13B3  4473          MOVZWL  CSB$W_CSID_IDX(R3),R0   ; CSID slot index
             3C    13           13B7  4474          BEQL    90$                     ; Branch if 0 -- invalid index
00000000'GF    50    B1        13B9  4475          CMPW    R0,G^CLU$GW_MAXINDEX    ; Valid index
             33    1E           13C0  4476          BGEQU   90$                     ; Branch if invalid index
51   00000000'GF    D0        13C2  4477          MOVL    G^CLU$GL_CLUSVEC,R1     ; Address of cluster vector
          6140     D5           13C9  4478          TSTL    (R1)[R0]                ; Is slot free?
             27    19           13CC  4479          BLSS    90$                     ; Branch if slot in use
       6140     53    D0        13CE  4480          MOVL    R3,(R1)[R0]             ; Store CSB address in slot
       54    64 A3    D0        13D2  4481          MOVL    CSB$L_CLUB(R3),R4       ; Address of CLUB
    05 60 A3    08    E0        13D6  4482          BBS     #CSB$V_CLUSTER, -       ; Skip ref time update if remote
                                13DB  4483                  CSB$L_STATUS(R3),5$     ;   node is already a cluster member
    74 A3    3C A4    7D        13DB  4484          MOVQ    CLUB$Q_LST_TIME(R4), -  ; Update reference time
                                13E0  4485                  CSB$Q_REFTIME(R3)
    05 60 A3    18    E1        13EC  4486 5$:      BBC     #CSB$V_LOCAL, -         ; Branch if not the local node
                                13E5  4487                  CSB$L_STATUS(R3),10$
    60 A4    4C A3    D0        13E5  4488          MOVL    CSB$L_CSID(R3), -       ; Store local CSID in CLUB
                                13EA  4489                  CLUB$L_LOCAL_CSID(R4)
    05 60 A3    00    E1        13EA  4490 10$:     BBC     #CSB$V_LONG_BREAK, -    ; Branch if no long break seen
                                13EF  4491                  CSB$L_STATUS(R3),20$
    00 1C A4    17    E2        13EF  4492          BBSS    #CLUB$V_LOST_CNX, -     ; Mark cluster connection lost
```

```
                                E 5
```

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

```
                                                13F9  4498          .SBTTL  REMOVE_NODE - Remove a node from the cluster
                                                13F9  4499 ;++
                                                13F9  4500 ;
                                                13F9  4501 ; FUNCTIONAL DESCRIPTION:
                                                13F9  4502 ;
                                                13F9  4503 ;       Remove the specified node from the cluster
                                                13F9  4504 ;
                                                13F9  4505 ; CALLING SEQUENCE:
                                                13F9  4506 ;
                                                13F9  4507 ;       JSB     REMOVE_NODE
                                                13F9  4508 ;       IPL is IPL$_SCS
                                                13F9  4509 ;
                                                13F9  4510 ; INPUT PARAMETERS:
                                                13F9  4511 ;
                                                13F9  4512 ;       R3:     CSB of the node to be removed
                                                13F9  4513 ;
                                                13F9  4514 ; OUTPUT PARAMETERS:
                                                13F9  4515 ;
                                                13F9  4516 ;       R3:     If CSB is deleted, the contents of back pointer cell.
                                                13F9  4517 ;               If CSB is not deleted, R3 is preserved.
                                                13F9  4518 ;
                                                13F9  4519 ; COMPLETION CODES:
                                                13F9  4520 ;
                                                13F9  4521 ;       NONE
                                                13F9  4522 ;
                                                13F9  4523 ; SIDE EFFECTS:
                                                13F9  4524 ;
                                                13F9  4525 ;       The cluster vector is updated.
                                                13F9  4526 ;       The CSB reference count is decremented.
                                                13F9  4527 ;       R0-R2 are destroyed.
                                                13F9  4528 ;
                                                13F9  4529 ;--
                                                13F9  4530
                                                13F9  4531 REMOVE_NODE:
                                      30   BB   13F9  4532          PUSHR   #^M<R4,R5>              ; Save register
                                 55   53   D0   13FB  4533          MOVL    R3,R5                   ; Address of CSB
                                50   0000'CF   9E   13FE  4534          MOVAB   W^FAILOVER_MSG,R0       ; Message for removing node from cluster
                                        EBFA'   30   1403  4535          BSBW    CNX$CONFIG_CHANGE       ; Tell the world the node is gone
                             50 60 A3   01   E5   1406  4536          BBCC    #CSB$V_MEMBER, -        ; Mark the node not a cluster member
                                                1408  4537                  CSB$L_STATUS(R3), 90$   ; *** Is this right?
                                60 A3   04   C8   1408  4538          BISL2   #CSB$M_REMOVED, -       ; Mark the node removed
                                                140F  4539                  CSB$L_STATUS(R3)
                             74 A3   3C A4   7D   140F  4540          MOVQ    CLUB$Q_LST_TIME(R4), -  ; Update reference time
                                                1414  4541                  CSB$Q_REFTIME(R3)
                                54   64 A3   D0   1414  4542          MOVL    CSB$L_CLUB(R3),R4       ; Fetch address of CLUB
                                                1418  4543 ;
                                                1418  4544 ; Tell quorum disk manager to skip a read cycle since a node has just been dropped
                                                1418  4545 ; from the cluster -- the removed node may still have a chance to write.
                                                1418  4546 ;
                          1C A4   01000000 8F   C8   1418  4547          BISL2   #CLUB$M_QF_FAILED_NODE, - ; Set bit requesting that a read cycle be
                                                1420  4548                  CLUB$L_FLAGS(R4)        ;   skipped
                          1C A4   40000000 8F   CA   1420  4549          BICL2   #CLUB$M_QF_DYNVOTE, -   ; Stop dynamically counting the quorum disk
                                                1428  4550                  CLUB$L_FLAGS(R4)        ;   until remote node can sense us
                                50   4C A3   3C   1428  4551          MOVZWL  CSB$W_CSID_IDX(R3),R0   ; CSID slot index
                                        2D   13   142C  4552          BEQL    90$                     ; Branch if 0 -- invalid index
                             00000000'GF   50   B1   142E  4553          CMPW    R0,G^CLU$GW_MAXINDEX    ; Valid index
                                        24   1E   1435  4554          BGEQU   90$                     ; Branch if invalid index
```

```
51   00000000'GF   D0  1437  4555          MOVL    G^CLU$GL_CLUSVEC,R1     ; Address of cluster vector
       53   6140   D1  143E  4556          CMPL    (R1)[R0],R3             ; Is slot consistent with CSB?
            17     12  1442  4557          BNEQ    90$                     ; Branch if things are inconsistent
     6140   4E A3  3C  1444  4558          MOVZWL  CSB$W_CSID_SEQ(R3), -   ; Store old sequence number for reuse
                       1449  4559                  (R1)[R0]
         55   53   D0  1449  4560          MOVL    R3,R5                   ; Address of CSB
            EBB1'  30  144C  4561          BSBW    CNX$DISC_REMOVE         ; Break connection to node that has been rem
            EBAE'  30  144F  4562          BSBW    SEND_JBCMSG             ; Tell the job controller that the node is g
            EBAB'  30  1452  4563          BSBW    CNX$DECREFCNT           ; Decrement reference count, perhaps delete
         53   55   D0  1455  4564          MOVL    R5,R3                   ; Address of this/previous CSB
              30   BA  1458  4565          POPR    #^M<R4,R5>              ; Restore R4 and R5
                   05  145A  4566          RSB
                       145B  4567
                       145B  4568 90$:     BUG_CHECK       CNXMGRERR,FATAL ; Consistency check
                       145F  4569
                       145F  4570
                       145F  4571          .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$_CLOEXIT                      ********    X   03
BUG$_CNXMGRERR                    ********    X   03
BUG$_OPERATOR                     ********    X   03
BUILD_STS_MSG                     000002C9  R     03
CDRP$C_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_NEWGDVOTES               = 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$INIT_SYD_RESP               00001309  RG  03
CNX$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$_TIMER                    = 00000C08
JOIN_CLUSTER                    00000887 R      03
JOIN_FINISH                     00000A86 R      03
JOIN_LOCKED                     00000974 R      03
JOIN_MSG                        ********    X   03
JOIN_PROPOSE                    00000A4F R      03
LCK$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
_255DUA28:[SYSLOA.OBJ]CLUSTER.MLB;1	2
_255DUA28:[SYS.OBJ]LIB.MLB;1	17
_255DUA28:[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