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

```
  CCCCCCCC  NN      NN  XX      XX    000000    PPPPPPPP    TTTTTTTTTT
  CCCCCCCC  NN      NN  XX      XX    000000    PPPPPPPP    TTTTTTTTTT
CC          NN      NN  XX      XX  00      00  PP      PP      TT
CC          NN      NN  XX      XX  00      00  PP      PP      TT
CC          NNNN    NN    XX  XX    00      00  PP      PP      TT
CC          NNNN    NN    XX  XX    00      00  PP      PP      TT
CC          NN  NN  NN      XX      00      00  PPPPPPPP        TT
CC          NN  NN  NN      XX      00      00  PPPPPPPP        TT
CC          NN    NNNN    XX  XX    00      00  PP              TT
CC          NN    NNNN    XX  XX    00      00  PP              TT
CC          NN      NN  XX      XX  00      00  PP              TT          ....
CC          NN      NN  XX      XX  00      00  PP              TT          ....
  CCCCCCCC  NN      NN  XX      XX    000000    PP              TT          ....
  CCCCCCCC  NN      NN  XX      XX    000000    PP              TT          ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

Table of contents

```
0000     1          .TITLE  CNXOPT - Optimal Subcluster Computation
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 contains the routines that compute an optimal fully-
0000    33 ;       connected subcluster of a given set of nodes.
0000    34 ;
0000    35 ; ENVIRONMENT: VAX/VMS
0000    36 ;
0000    37 ; AUTHOR: Dave Thiel,    CREATION DATE: 09-Dec-1983
0000    38 ;
0000    39 ; MODIFIED BY:
0000    40 ;
0000    41 ;       V03-005 DWT0223         David W. Thiel                  27-Jun-1984
0000    42 ;               Add debugging traps.  Correct calculation to do consistent
0000    43 ;               bookkeeping.
0000    44 ;
0000    45 ;       V03-004 DWT0185         David W. Thiel                  2-Mar-1984
0000    46 ;               Fix blown register.
0000    47 ;
0000    48 ;       V03-003 DWT0175         David W. Thiel                  21-Feb-1984
0000    49 ;               Minimize quorum disk votes against value in CLUB.
0000    50 ;
0000    51 ;       V03-002 DWT0165         David W. Thiel                  08-Feb-1984
0000    52 ;               Set up R3 before calling REMOVE_AMAP at 230$.
0000    53 ;
0000    54 ;       V03-001 DWT0162         David W. Thiel                  01-Feb-1984
0000    55 ;               Add CNX$OPT_INIT entry point.  Add management of quorum
0000    56 ;               disk bit in CSB$B_NODEMAP.  Correct coding errors present
0000    57 ;               in first pass.
```

```
0000     58 ;
0000     59 ;--
0000     60
```

```
                        .SBTTL  DECLARATIONS
                ;
                ; INCLUDE FILES:
                ;
                        $CLUBDEF                        ; CLUster Block offsets
                        $CLUOPTDEF                      ; CLUster OPTimization block offsets
                        $CSBDEF                         ; CSB Offsets
                        $DYNDEF                         ; Data structure type codes
                        $FKBDEF                         ; Fork block offsets

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

                ;
                ; Assume that all bitmaps are of the same size
                ;
                        ASSUME  CLUOPT$S_CMAP,EQ,CLUOPT$S_AMAP
                        ASSUME  CLUOPT$S_CMAP,EQ,CLUOPT$S_RMAP
                        ASSUME  CLUOPT$S_CMAP,EQ,CSB$S_NODEMAP
                        ASSUME  CLUOPT$S_CMAP,EQ,CLUB$S_NODEMAP

                        .DEFAULT        DISPLACEMENT,WORD

00000000                .PSECT  $$$100,LONG             ; PSECT for code
```

```
                                0000    91          .SBTTL  CNX$OPT_INIT - Compute Optimal Initial Subcluster
                                0000    92
                                0000    93 ;++
                                0000    94 ;
                                0000    95 ; FUNCTIONAL DESCRIPTION:
                                0000    96 ;
                                0000    97 ;       This routine is is called to compute the optimal completely
                                0000    98 ;       interconnected subcluster of the nodes marked with the select bit.
                                0000    99 ;       No previous suggested subcluster is assumed.
                                0000   100 ;
                                0000   101 ; CALLING SEQUENCE:
                                0000   102 ;
                                0000   103 ;       JSB     CNX$OPT_INIT
                                0000   104 ;       IPL is IPL$_SCS=IPL$_SYNCH
                                0000   105 ;
                                0000   106 ; INPUT PARAMETERS:
                                0000   107 ;
                                0000   108 ;       For all CSB's with the SELECTED bit bit:
                                0000   109 ;           CSB$B_NODEMAP is a (potentially optimistic) estimate of the
                                0000   110 ;           connectivity of the node.
                                0000   111 ;       In both NODEMAP's, bit 0 is used to indicate the state of the
                                0000   112 ;       "connection" to the quorum disk.  This bit should be set only
                                0000   113 ;       if the quorum disk is the same as on the executing node and
                                0000   114 ;       the disk/quorum file is accessible to the subject node.
                                0000   115 ;
                                0000   116 ; OUTPUT PARAMETERS:
                                0000   117 ;
                                0000   118 ;       CLUB$L_FMERIT is the figure of merit of the computed cluster.
                                0000   119 ;       CLUB$B_NODEMAP describes the members of the computed cluster.
                                0000   120 ;               Bit 0 indicates quorum disk participation.
                                0000   121 ;
                                0000   122 ; COMPLETION CODES:
                                0000   123 ;
                                0000   124 ;       R0 contains status.
                                0000   125 ;       If R0 indicates success, R1 will always contain TRUE
                                0000   126 ;
                                0000   127 ; SIDE EFFECTS:
                                0000   128 ;
                                0000   129 ;       NONE
                                0000   130 ;
                                0000   131 ;--
                                0000   132
                                0000   133 CNX$OPT_INIT::
                          3C    BB 0000   134           PUSHR   #^M<R2,R3,R4,R5>        ; Save registers
          54    00000000'GF   D0 0002   135           MOVL    G^CLU$GL_CLUB,R4        ; Fetch address of CLUB
                      00A8 C4   D4 0009   136           CLRL    CLUB$L_FMERIT(R4)       ; Clear previous figure of merit
00EC C4   20    00    6E    00  2C 000D   137           MOVC5   #0,(SP),#0, -           ; Zero previous description
                                0015   138                   #CLUB$S_NODEMAP, -
                                0015   139                   CLUB$B_NODEMAP(R4)
                          3C    BA 0015   140           POPR    #^M<R2,R3,R4,R5>        ; Restore registers
                                0017   141 ;         BRB     CNX$OPT                 ; Fall into CNX$OPT
                                0017   142
```

```
0017   144          .SBTTL  CNX$OPT - Compute Optimal Subcluster
0017   145
0017   146 ;++
0017   147 ;
0017   148 ; FUNCTIONAL DESCRIPTION:
0017   149 ;
0017   150 ;       This routine is is called to compute the optimal completely
0017   151 ;       interconnected subcluster of the nodes marked with the select bit.
0017   152 ;
0017   153 ; CALLING SEQUENCE:
0017   154 ;
0017   155 ;       JSB     CNX$OPT
0017   156 ;       IPL is IPL$_SCS=IPL$_SYNCH
0017   157 ;
0017   158 ; INPUT PARAMETERS:
0017   159 ;
0017   160 ;       CLUB$B_NODEMAP is an initial cluster to try to better.
0017   161 ;       CLUB$L_FMERIT is the figure of merit of the initial cluster.
0017   162 ;       For all CSB's with the SELECTED bit bit:
0017   163 ;           CSB$B_NODEMAP is a (potentially optimistic) estimate of the
0017   164 ;           connectivity of the node.
0017   165 ;       In both NODEMAP's, bit 0 is used to indicate the state of the
0017   166 ;       "connection" to the quorum disk.  This bit should be set only
0017   167 ;       if the quorum disk is the same as on the executing node and
0017   168 ;       the disk/quorum file is accessible to the subject node.
0017   169 ;
0017   170 ; OUTPUT PARAMETERS:
0017   171 ;
0017   172 ;       CLUB$L_FMERIT is the figure of merit of the computed cluster.
0017   173 ;       CLUB$B_NODEMAP describes the members of the computed cluster.
0017   174 ;               Bit 0 indicates quorum disk participation.
0017   175 ;
0017   176 ; COMPLETION CODES:
0017   177 ;
0017   178 ;       R0 contains status.
0017   179 ;       If R0 indicates success, R1 indicates whether an improved cluster
0017   180 ;               (improved over the initial cluster described in CLUB$L_FMERIT
0017   181 ;                and CLUB$B_NODEMAP) was found.
0017   182 ;
0017   183 ; SIDE EFFECTS:
0017   184 ;
0017   185 ;       NONE
0017   186 ;
0017   187 ; DESCRIPTION:
0017   188 ;
0017   189 ;       This procedure investigates all possible fully connected subclusters
0017   190 ;       that include the local node and chooses the one with the largest figure
0017   191 ;       of merit.  The figure of merit is defined as:
0017   192 ;               (sum of the votes of the nodes * 256) + number of nodes
0017   193 ;       The votes of the quorum disk are counted, but the quorum disk as a
0017   194 ;       node is not counted.
0017   195 ;
0017   196 ;       The search is done recursively, using a linked list of CLUOPT structures
0017   197 ;       as the stack for the recursion.  The CLUOPT structure contains the
0017   198 ;       following interesting fields:
0017   199 ;       CLUOPT$L_PREV:  Link to previous block
0017   200 ;       CLUOPT$L_CMERIT: Sum of the votes of the VAX nodes in CMAP
```

```
                     0017   201 ;       CLUOPT$L_ACMERIT: Sum of the votes of the VAX nodes in CMAP and AMAP
                     0017   202 ;       CLUOPT$B_CMAP:  Bit map of nodes already included in the cluster being
                     0017   203 ;                       computed.  All of the nodes in CMAP are fully
                     0017   204 ;                       interconnected.
                     0017   205 ;       CLUOPT$B_AMAP:  Bit map of nodes available for inclusion in the cluster
                     0017   206 ;                       being computed.
                     0017   207 ;       CLUOPT$B_RMAP:  Bit map of nodes excluded from consideration in the
                     0017   208 ;                       cluster by virtue of at least one node in CMAP not
                     0017   209 ;                       having a connection to the node.
                     0017   210 ;
                     0017   211 ;       CLUB$B_NODEMAP and CLUB$L_FMERIT are updated every time a better cluster
                     0017   212 ;       is found.
                     0017   213 ;
                     0017   214 ;       A direct implemention of the search required takes (N-1)! (factorial) steps.
                     0017   215 ;       The execution time is kept within reason for the expected cases by the
                     0017   216 ;       following heuristic techniques:
                     0017   217 ;       1.  Whenever the upper bound on what may be attainable is worse than what
                     0017   218 ;           has already been achieved, the recursion is abandoned.
                     0017   219 ;       2.  Whenever multiple nodes are equivalent, they are dealt with
                     0017   220 ;           simultaneously, reducing the breadth and depth of the search.
                     0017   221 ;
                     0017   222 ;--
                     0017   223
                     0017   224 CNX$OPT::
           07FC 8F   BB 0017   225         PUSHR   #^M<R2,R3,R4,R5,R6,R7,R8,R9,R10>        ; Save some registers
                7E   D4 001B   226         CLRL    -(SP)                           ; Flag indicating no improved cluster found
                59   7C 001D   227         CLRQ    R9                              ; R9 is top of frame stack
                        001F   228                                                 ; R10 is address of mapping vector
      51   040C 8F   3C 001F   229         MOVZWL  #<<CSB$S_NODEMAP*8>*4>+12, -    ; One longword per possible system
                        0024   230                 R1                              ;   plus standard header
              FFD9'   30 0024   231         BSBW    CNX$ALLOZMEM                    ; Allocate and zero memory
                15 50   E9 0027   232         BLBC    R0,10$                          ; Branch on error
                5A   52   D0 002A   233         MOVL    R2,R10                          ; Address of mapping vector
           50   FF 8F   9A 002D   234         MOVZBL  #<CSB$S_NODEMAP*8>-1,R0         ; number of phoney cells
        0C AA40   01   D0 0031   235 5$:       MOVL    #1,12(R10)[R0]                  ; catch use of uninitialized cell
                F8 50   F5 0036   236         SOBGTR  R0,5$
           0B AA   02   90 0039   237         MOVB    #DYN$C_CLU_CLUVEC, -            ; Block subtype
                        003D   238                 FKB$B_TYPE+1(R10)               ; Use block type of cluster vector
                        003D   239 ;
                        003D   240 ; This call enters the main body of the routine.
                        003D   241 ; It exists only so that several points in the routine can get
                        003D   242 ; to the common exit with an RSB instruction.
                        003D   243 ;
                33   10 003D   244         BSBB    100$                            ; Call main section
                50   DD 003F   245 10$:    PUSHL   R0                              ; Save return status
           50   59   D0 0041   246 20$:    MOVL    R9,R0                           ; Stack frame address
                0B   13 0044   247         BEQL    30$                             ; Branch if no frame
           59   69   D0 0046   248         MOVL    CLUOPT$L_PREV(R9),R9            ; Pop the stack
     00000000'GF   16 0049   249         JSB     G^EXE$DEANONPAGED               ; Deallocate the frame
                F0   11 004F   250         BRB     20$                             ; Iterate to flush more frames
                        0051   251
           50   5A   D0 0051   252 30$:    MOVL    R10,R0                          ; Vector address
                06   13 0054   253         BEQL    40$                             ; Branch if no
     00000000'GF   16 0056   254         JSB     G^EXE$DEANONPAGED               ; Deallocate vector
                        005C   255 40$:
              FFA1'   30 005C   256         BSBW    CNX$SCAN_CSBS                   ; Iterate over all CSBs
                06 50   E9 005F   257         BLBC    R0,50$                          ; Branch when done
```

```
                008C C3   01    8A  0062   258          BICB    #1,CSB$B_NODEMAP(R3)    ; Clear quorum disk bit
                          05    0067   259          RSB
                                0068   260
                00EC C4   01    8A  0068   261 50$:     BICB    #1,CLUB$B_NODEMAP(R4)   ; Clear quorum disk bit in CLUB
                     07FF 8F    BA  006D   262          POPR    #^M<R0,R1,R2,R3,R4,R5,R6,R7,R8,R9,R10>  ; Restore registers, fetch s
                          05    0071   263 60$:     RSB                             ; Best cluster is in CLUB$B_NODEMAP
                                0072   264 ;
                                0072   265 ; Main entrance to the optimal cluster allocation.
                                0072   266 ; The first thing to do is to allocate the CLUOPT block that describes the basis
                                0072   267 ; of the recursion.  Then, the selected nodes as placed into AMAP and into the
                                0072   268 ; vector (R10) that maps bit numbers into CSB addresses.  The local node is the
                                0072   269 ; only initial member of CMAP.  This initializes the recursion to start from a
                                0072   270 ; point where the local node must be a member of any computed cluster and all other
                                0072   271 ; selected nodes are potential cluster members.
                                0072   272 ;
                  51    74 8F   9A  0072   273 100$:     MOVZBL  #CLUOPT$K_LENGTH,R1     ; Length
                        FF87'   30  0076   274          BSBW    CNX$ALLOZMEM            ; Allocate and zero memory
                      F5 50     E9  0079   275          BLBC    R0,60$                  ; Branch on error
                   59    52     D0  007C   276          MOVL    R2,R9                   ; Update stack frame
                0B A9    06     90  007F   277          MOVB    #DYN$C_CLU_CLUOPT, -    ; Block sub-type
                                0083   278                  CLUOPT$B_SUBTYPE(R9)
                      FF7A'     30  0083   279          BSBW    CNX$SCAN_CSBS           ; iterate over all CSBs
                    45 50       E9  0086   280          BLBC    R0,140$                 ; Branch when done
             3B 60 A3    11     E1  0089   281          BBC     #CSB$V_SELECTED, -      ; Branch if not selected
                                008E   282                  CSB$L_STATUS(R3),130$
              008C C3   01      8A  008E   283          BICB    #1,CSB$B_NODEMAP(R3)    ; Clear quorum disk bit
             0A 60 A3   03      E1  0093   284          BBC     #CSB$V_QF_SAME, -       ; Branch if quorum disk not
                                0098   285                  CSB$L_STATUS(R3),110$   ;   same as local node
             05 60 A3   09      E1  0098   286          BBC     #CSB$V_QF_ACTIVE, -     ; Branch if quorum disk not
                                009D   287                  CSB$L_STATUS(R3),110$   ;   accessible
              008C C3   01      88  009D   288          BISB    #1,CSB$B_NODEMAP(R3)    ; Mark connection to quorum disk
                51    4C A3    3C  00A2   289 110$:     MOVZWL  CSB$W_CSID_IDX(R3),R1   ; CSID index
                 0C AA41   53    D0  00A6   290          MOVL    R3,12(R10)[R1]          ; Store CSB address in vector
             1A 34 A9     51    E2  00AB   291          BBSS    R1,CLUOPT$B_AMAP(R9),135$ ; Set bit in available map
                50    50 A3    3C  00B0   292 120$:     MOVZWL  CSB$W_VOTES(R3),R0      ; Votes held by node
                 50    50   08    78  00B4   293          ASHL    #8,R0,R0                ; Scale votes
                           50    D6  00B8   294          INCL    R0                      ; Count the node
                   10 A9    50    C0  00BA   295          ADDL2   R0,CLUOPT$L_ACMERIT(R9) ; Update ACMERIT
             06 60 A3    18    E1  00BE   296          BBC     #CSB$V_LOCAL, -         ; Branch if not local CSB
                                00C3   297                  CSB$L_STATUS(R3),130$
                      52    59  D0  00C3   298          MOVL    R9,R2                   ; Stack frame for ADD_CMAP
                         010D   31  00C6   299          BRW     ADD_CMAP                ; Add the node to CMAP and return
                                05  00C9   300 130$:     RSB
                                00CA   301
                                00CA   302 135$:     BUG_CHECK       CNXMGRERR,FATAL ; funny data
                                00CE   303
             F7 34 A9    00    E2  00CE   304 140$:     BBSS    #0,CLUOPT$B_AMAP(R9),135$ ; Mark the quorum disk available
                                00D3   305 ;
                                00D3   306 ; This is the entry point into the recursion.
                                00D3   307 ; R9 is the address of the CLUOPT block for the current level of recursion.
                                00D3   308 ;       CMAP(R9) is the map of nodes definitely in the cluster being computed.
                                00D3   309 ;       AMAP(R9) is the map of nodes that are still candidates for inclusion
                                00D3   310 ;                   in the cluster being computed.
                                00D3   311 ;       PREV(R9) is the address of the previous CLUOPT block in the recursion.
                                00D3   312 ;       CMERIT(R9) is the figure of merit of the VAX nodes in CMAP.
                                00D3   313 ;       ACMERIT(R9) is the figure of merit of all of the VAX nodes in AMAP and
                                00D3   314 ;                   CMAP.  It is thus an upper bound on the figure of merit
```

```
                                00D3    315 ;                           of any possible cluster, exclusive of the contribution
                                00D3    316 ;                           of a possible quorum disk.
                                00D3    317 ; R10 is the address of the CSB vector
                                00D3    318 ;
                                00D3    319 200$:
                                00D3    320 ;
                                00D3    321 ; Remove from AMAP every node not connected to all nodes in CMAP
                                00D3    322 ;
                      34 A9     9F    00D3    323          PUSHAB  CLUOPT$B_AMAP(R9)       ; Address of map to scan
                       0170     30    00D6    324          BSBW    SCAN_MAP                ; Initializer map scanner
                      26 50     E9    00D9    325          BLBC    R0,240$                 ; Branch when done
                 53   0C AA41   D0    00DC    326          MOVL    12(R10)[R1],R3          ; CSB address
               16 54 A9    51   E0    00E1    327          BBS     R1,CLUOPT$B_RMAP(R9),230$ ; Some node in CMAP is not connected to no
                           53   D5    00E6    328          TSTL    R3                      ; Nonexistent CSB? (quorum disk)
                           11   13    00E8    329          BEQL    220$                    ; Ignore quorum disk
                                00EA    330          ASSUME  CLUOPT$S_CMAP&3,EQ,0    ; Assume bitmap is an integral number of lon
                                00EA    331          ASSUME  CLUOPT$S_CMAP,GE,4      ; Assume at least one iteration
                      50    07    D0    00EA    332          MOVL    #<CLUOPT$S_CMAP/4>-1,R0
 52   14 A940    008C C340    CB    00ED    333 210$:    BICL3   CSB$B_NODEMAP(R3)[R0], - ; Look for missing connection to any node i
                                00F6    334                  CLUOPT$B_CMAP(R9)[R0],R2 ;  CMAP
                           04   12    00F6    335          BNEQ    230$                    ; Branch if connection is missing
                      F2 50     F4    00F8    336          SOBGEQ  R0,210$                 ; Iterate over entire connection map
                           05   05    00FB    337 220$:    RSB                             ; Return to scanner
                                00FC    338
                      52   59   D0    00FC    339 230$:    MOVL    R9,R2                   ; Stack frame for removal
                       0107    31    00FF    340          BRW     REMOVE_AMAP             ; Remove from AMAP and return to scanner
                               0102    341
                               0102    342 240$:
                               0102    343 ;
                               0102    344 ; Loop at a given recursion depth.
                               0102    345 ; R9 is CLUOPT block for this depth.
                               0102    346 ; R10 is vector of CSB's.
                               0102    347 ;
                               0102    348 300$:
                               0102    349 ;
                               0102    350 ; Determine whether the recursion can be discontinued because the best possible resu
                               0102    351 ; is not better than the best result already attained.
                               0102    352 ; Compute an upper bound the the figure of merit by summing the votes of all nodes
                               0102    353 ; in AMAP and CMAP with the minimum of the votes proposed for the quorum disk by
                               0102    354 ; the nodes in CMAP.
                               0102    355 ;
          54   00000000'GF     D0    0102    356          MOVL    G^CLU$GL_CLUB,R4        ; Address of CLUB
                57   10 A9     D0    0109    357          MOVL    CLUOPT$L_ACMERIT(R9),R7 ; Contribution of VAX nodes
                     06 54 A9  E8    010D    358          BLBS    CLUOPT$B_RMAP(R9),310$  ; Branch if quorum disk is excluded
                        0111   30    0111    359          BSBW    QDMERIT                 ; Calculate quorum disk contribution
                     57   50   C0    0114    360          ADDL2   R0,R7                   ; Include contribution of quorum disk
              00A8 C4    57   D1    0117    361 310$:    CMPL    R7,CLUB$L_FMERIT(R4)    ; Compare best attainable to best already
                           35   1B    011C    362          BLEQU   340$                    ; Branch if no improvement possible
                                011E    363 ;
                                011E    364 ; Pick a node from AMAP
                                011E    365 ; If none are available, this recursion level is done
                                011E    366 ;
                                011E    367          ASSUME  CLUOPT$S_AMAP&3,EQ,0
                                011E    368          ASSUME  CLUOPT$S_AMAP,GE,4      ; Assume at least one iteration
                      51   08   D0    011E    369          MOVL    #CLUOPT$S_AMAP/4,R1     ; Number of iterations
                           50   D4    0121    370          CLRL    R0                      ; Starting bit position
 50   34 A9     20     50   EA    0123    371 320$:    FFS     R0,#32,CLUOPT$B_AMAP(R9),R0
```

```
                     3C   12  0129   372          BNEQ    400$                    ; Branch if bit found
                  F5 51   F5  012B   373          SOBGTR  R1,320$                 ; Iterate over all longwords
                          012E   374 ;
                          012E   375 ; No nodes in list of remaining nodes.
                          012E   376 ; We are at the bottom of the recursion.
                          012E   377 ; Check for best cluster yet and store new best cluster.
                          012E   378 ;
               57   0C A9   D0  012E   379          MOVL    CLUOPT$L_CMERIT(R9),R7  ; VAX contribution to figure of merit
                 06 14 A9   E9  0132   380          BLBC    CLUOPT$B_CMAP(R9),330$  ; Branch if quorum disk not included
                    00EC    30  0136   381          BSBW    QDMERIT                 ; Calculate disk contribution
                       57   50  C0  0139   382          ADDL2   R0,R7
            00A8 C4    57   D1  013C   383 330$:    CMPL    R7,CLUB$L_FMERIT(R4)    ; Compare this cluster to best seen so far
                       10   1B  0141   384          BLEQU   340$                    ; Branch if new one is no better, return
            00A8 C4    57   D0  0143   385          MOVL    R7,CLUB$L_FMERIT(R4)    ; Save figure of merit
      00EC C4   14 A9   20   28  0148   386          MOVC3   #CLUB$S_NODEMAP, -
                                014F   387                  CLUOPT$B_CMAP(R9), -
                                014F   388                  CLUB$B_NODEMAP(R4)
               04 AE   00'   D0  014F   389          MOVL    S^#SS$_NORMAL,4(SP)     ; Mark an improved cluster found
                  50   59   D0  0153   390 340$:    MOVL    R9,R0                   ; Address of CLUOPT block
                  59   69   D0  0156   391          MOVL    CLUOPT$L_PREV(R9),R9    ; POP CLUOPT block stack
            00000000'GF    16  0159   392          JSB     G^EXE$DEANONPAGED       ; Deallocate this CLUOPT block
                       59   D5  015F   393          TSTL    R9                      ; End of stack?
                       9F   12  0161   394          BNEQ    300$                    ; Branch if no to continue
                            0163   395 ;
                            0163   396 ; The recursion is complete.  This is the main exit.
                            0163   397 ;
                  50   00'   D0  0163   398          MOVL    S^#SS$_NORMAL,R0        ; Successful return
                       05   0166   399 350$:    RSB
                            0167   400 ;
                            0167   401 ; A node (index in R0) has been chosen as the basis for constructing a new frame
                            0167   402 ; and taking the recursion down a level.
                            0167   403 ; Begin building a new frame.
                            0167   404 ;
            53   0C AA40   D0  0167   405 400$:    MOVL    12(R10)[R0],R3          ; CSB of basis node
                            016C   406 ;
                            016C   407 ; Regsister/Data available:
                            016C   408 ;
                            016C   409 ;       0(SP)   CSB for chosen basis node
                            016C   410 ;       R8:     CLUOPT for previous frame
                            016C   411 ;       R9:     CLUOPT for new current frame
                            016C   412 ;       R10:    Vector of nodes
                            016C   413 ;
              51   74 8F   9A  016C   414          MOVZBL  #CLUOPT$K_LENGTH,R1     ; Length
            00000000'GF    16  0170   415          JSB     G^EXE$ALONONPAGED       ; Allocate memory
                  ED 50   E9  0176   416          BLBC    R0,350$                 ; Branch on error
                  58   59   D0  0179   417          MOVL    R9,R8                   ; New previous frame
                  59   52   D0  017C   418          MOVL    R2,R9                   ; New current frame
                       0A   BB  017F   419          PUSHR   #^M<R1,R3>              ; Save CSB address and frame length
69 51 00 68   08 A8   2C  0181   420          MOVC5   CLUOPT$W_SIZE(R8),(R8), - ; Copy old block into the new
                                0188   421                  #0,R1,(R9)
               08 A9   8E   F7  0188   422          CVTLW   (SP)+,CLUOPT$W_SIZE(R9) ; Restore size word
                  69   58   D0  018C   423          MOVL    R8,CLUOPT$L_PREV(R9)    ; Link to previous frame
                            018F   424 ;
                            018F   425 ; Regsister/Data available:
                            018F   426 ;
                            018F   427 ;       0(SP)   CSB for chosen addition
                            018F   428 ;       R8:     CLUOPT for previous frame
```

```
                                018F    429 ;       R9:     CLUOPT for new current frame
                                018F    430 ;       R10:    Vector of nodes
                                018F    431 ;
                                018F    432 ; Find equivalent nodes in set of remaining nodes (AMAP).
                                018F    433 ; Nodes are equivalent if they see the same connectivity with respect to
                                018F    434 ; cluster and available nodes that are seen by the chosen node.
                                018F    435 ; Note that the chosen basis node is added by virtue of it being equivalent
                                018F    436 ; to itself.  The quorum disk is treated as a special case because no CSB
                                018F    437 ; exists for the disk proper.
                                018F    438 ;
                          54 8ED0   018F    439         POPL    R4                      ; Is the chosen node the quorum disk?
                          39   13   0192    440         BEQL    450$                    ; Branch if yes
                    34 A8    9F   0194    441         PUSHAB  CLUOPT$B_AMAP(R8)       ; Scan available nodes in previous frame
                     00AF    30   0197    442         BSBW    SCAN_MAP                ; Initialize scanner
                    36 50    E9   019A    443         BLBC    R0,460$                 ; Branch when done
               53   0C AA41  D0   019D    444         MOVL    12(R10)[R1],R3          ; CSB of node under consideration
                         28   13   01A2    445         BEQL    440$                    ; Branch if quorum disk and ignore it
                              01A4    446         ASSUME  CSB$S_NODEMAP&3,EQ,0    ; Assume an integral number of longwords
                              01A4    447         ASSUME  CSB$S_NODEMAP,GE,4      ; Assume at least one iteration
                         52   07   D0   01A4    448         MOVL    #<CSB$S_NODEMAP/4>-1,R2 ; Iterate over all bytes of map
55   008C C342   008C C442   CD   01A7    449 420$:   XORL3   CSB$B_NODEMAP(R4)[R2], - ; Compute differences between chosen
                              01B1    450                     CSB$B_NODEMAP(R3)[R2],R5 ;   and candidate nodes
                              01B1    451 ;
                              01B1    452 ; Check if any of the differences reflect nodes in either the new CMAP or new AMAP.
                              01B1    453 ; Note that new CMAP .OR. new AMAP is a constant during the execution of this phase
                              01B1    454 ; since the only changes allowed move bits from one to the other.
                              01B1    455 ;
               14 A942    55   D3   01B1    456         BITL    R5,CLUOPT$B_CMAP(R9)[R2] ; Check for differences with CMAP nodes con
                         14   12   01B6    457         BNEQ    440$                    ; Branch on discrepancy and reject this addi
               34 A942    55   D3   01B8    458         BITL    R5,CLUOPT$B_AMAP(R9)[R2] ; Check for differences with AMAP nodes' co
                         0D   12   01BD    459         BNEQ    440$                    ; Branch on discrepancy and reject this addi
                    E5 52    F4   01BF    460         SOBGEQ  R2,420$
                              01C2    461 ;
                              01C2    462 ; At this point, the node whose index is in R1 has the same connectivity, with respe
                              01C2    463 ; to nodes in CMAP and AMAP, as the basis node and is therefore equivalent to the ba
                              01C2    464 ; node in every respect.  The basis node plus all equivalent nodes are moved from AM
                              01C2    465 ; to CMAP simultaneously.  This is a major optimization of the direct implementation
                              01C2    466 ; of this search (which takes N! (N factial) steps) because the commonly anticipated
                              01C2    467 ; cases have large numbers of equivalent nodes.
                              01C2    468 ;
                              01C2    469 ; Note that a BSB call is made to this point to handle the quorum disk!
                              01C2    470 ;
52   59   D0   01C2    471 430$:   MOVL    R9,R2                   ; Stack frame for ADD_CMAP
     0F   10   01C5    472         BSBB    ADD_CMAP                ; Add node to new CMAP
52   58   D0   01C7    473         MOVL    R8,R2                   ; Use previous CLUOPT address
     3D   10   01CA    474         BSBB    REMOVE_AMAP             ; Remove from old AMAP and return
          05   01CC    475 440$:   RSB
               01CD    476
          51   D4   01CD    477 450$:   CLRL    R1                      ; Disk is node 0
          53   D4   01CF    478         CLRL    R3                      ; No CSB
          EF   10   01D1    479         BSBB    430$                    ; Use same code as for normal case
        FEFD   31   01D3    480 460$:   BRW     200$                    ; The new recursion is set up -- do it!
```

```
                                01D6   482          .SBTTL  ADD_CMAP - Add a node to CMAP
                                01D6   483
                                01D6   484 ;++
                                01D6   485 ;
                                01D6   486 ; FUNCTIONAL DESCRIPTION:
                                01D6   487 ;
                                01D6   488 ;       This routine adds a node to CMAP, adjusts the figure of merit for the
                                01D6   489 ;       subcluster attained in CMAP and the best potential subcluster, and
                                01D6   490 ;       updates RMAP to reflect any additional nodes whose membership is
                                01D6   491 ;       incompatibile with the subject node.
                                01D6   492 ;       If the subject node is a member of AMAP, it is removed.
                                01D6   493 ;
                                01D6   494 ; CALLING SEQUENCE:
                                01D6   495 ;
                                01D6   496 ;       BSBB/BSBW/JSB   ADD_CMAP
                                01D6   497 ;
                                01D6   498 ; INPUT PARAMETERS:
                                01D6   499 ;
                                01D6   500 ;       R1:             Index of the node to add to CMAP
                                01D6   501 ;       R2:             Address of CLUOPT block containing the CMAP and AMAP
                                01D6   502 ;       R3:             Address of the CSB of the node to add
                                01D6   503 ;
                                01D6   504 ; OUTPUT PARAMETERS:
                                01D6   505 ;
                                01D6   506 ;       NONE
                                01D6   507 ;
                                01D6   508 ; COMPLETION CODES:
                                01D6   509 ;
                                01D6   510 ;       NONE
                                01D6   511 ;
                                01D6   512 ; SIDE EFFECTS:
                                01D6   513 ;
                                01D6   514 ;       The contents of R0 are destroyed.
                                01D6   515 ;
                                01D6   516 ;--
                                01D6   517
                                01D6   518 ADD_CMAP:
2A 14 A2    51    E2            01D6   519          BBSS    R1,CLUOPT$B_CMAP(R2),30$ ; Add to new CMAP, branch if present
              51    D5            01DB   520          TSTL    R1                       ; Is the subject the quorum disk
                23    13            01DD   521          BEQL    20$                      ; Branch if quorum disk
   50    50 A3    3C            01DF   522          MOVZWL  CSB$W_VOTES(R3),R0       ; Votes held by node
 50    50    08    78            01E3   523          ASHL    #8,R0,R0                 ; Count each vote as 256 points
              50    D6            01E7   524          INCL    R0                       ; Include the node
   0C A2      50    C0            01E9   525          ADDL2   R0,CLUOPT$L_CMERIT(R2)   ; Add to CMERIT
   10 A2      50    C0            01ED   526          ADDL2   R0,CLUOPT$L_ACMERIT(R2)  ; Add to ACMERIT
                                01F1   527          ASSUME  CSB$S_NODEMAP&3,EQ,0     ; Assume map is an integral number of longwo
                                01F1   528          ASSUME  CSB$S_NODEMAP,GE,4       ; Assume at least one iteration
     50    07    D0            01F1   529          MOVL    #<CSB$S_NODEMAP/4>-1,R0  ; Update rejection map
7E    008C C340    D2            01F4   530 10$:     MCOML   CSB$B_NODEMAP(R3)[R0],-(SP) ; Nodes not connected to subject node
  54 A240    8E    C8            01FA   531          BISL2   (SP)+,CLUOPT$B_RMAP(R2)[R0] ; Mark unconnected nodes as rejected
        F2 50    F4            01FF   532          SOBGEQ  R0,10$                   ; Iterate over all longwords of map
              05    10            0202   533 20$:     BSBB    REMOVE_AMAP              ; Remove the node from AMAP
              05    0204   534          RSB                              ; Fall into REMOVE_AMAP
                                0205   535
                                0205   536 30$:     BUG_CHECK       CNXMGRERR,FATAL ; Invalid state
```

```
                                                                .SBTTL  REMOVE_AMAP - Remove a node from AMAP

                                                        ;++
                                                        ;
                                                        ; FUNCTIONAL DESCRIPTION:
                                                        ;
                                                        ;       This routine removes a node from AMAP and adjusts the figure of
                                                        ;       merit for the best potential subcluste to reflect the absence of
                                                        ;       this node.
                                                        ;
                                                        ; CALLING SEQUENCE:
                                                        ;
                                                        ;       BSBB/BSBW/JSB   REMOVE_AMAP
                                                        ;
                                                        ; INPUT PARAMETERS:
                                                        ;
                                                        ;       R1:             Index of the node to add to CMAP
                                                        ;       R2:             Address of CLUOPT block containing the CMAP and AMAP
                                                        ;       R3:             Address of the CSB of the node to add
                                                        ;
                                                        ; OUTPUT PARAMETERS:
                                                        ;
                                                        ;       NONE
                                                        ;
                                                        ; COMPLETION CODES:
                                                        ;
                                                        ;       NONE
                                                        ;
                                                        ; SIDE EFFECTS:
                                                        ;
                                                        ;       The contents of R0 are destroyed.
                                                        ;
                                                        ;--

                                                REMOVE_AMAP:
13 34 A2    51  E5                              BBCC    R1,CLUOPT$B_AMAP(R2),20$ ; Remove from AMAP, branch if not present
           51   D5                              TSTL    R1                       ; Is the subject the quorum disk?
           0E   13                              BEQL    10$                      ; Branch if yes
   50   50 A3   3C                              MOVZWL  CSB$W_VOTES(R3),R0       ; Votes held by node
50   50    08   78                              ASHL    #8,R0,R0                 ; Count each vote at 256 points
           50   D6                              INCL    R0                       ; Include the node
  10 A2    50   C2                              SUBL2   R0,CLUOPT$L_ACMERIT(R2) ; Remove from ACMERIT
                05                      10$:    RSB
                                                ;
                                        20$:    BUG_CHECK       CNXMGRERR,FATAL ; Invalid state
```

```
                     0225   584          .SBTTL  QDMERIT - Quorum disk contribution to figure of merit
                     0225   585
                     0225   586 ;++
                     0225   587 ;
                     0225   588 ; FUNCTIONAL DESCRIPTION:
                     0225   589 ;
                     0225   590 ;       This routine computes the contribution of the quorum disk to the figure
                     0225   591 ;       of merit.  It assumes that the quorum disk does contribute.  The
                     0225   592 ;       contribution is calculated as the minimum of the votes proposed for the
                     0225   593 ;       quorum disk by each of the nodes in CMAP.  This minimum is scaled in
                     0225   594 ;       the same way as the votes contributed by a VAX node.  However, the
                     0225   595 ;       quorum disk does not get another point to represent the node itself.
                     0225   596 ;       Thus a VAX node with 1 vote is more desirable than a 1 vote contribution
                     0225   597 ;       from the quorum disk.
                     0225   598 ;
                     0225   599 ; CALLING SEQUENCE:
                     0225   600 ;
                     0225   601 ;       BSBB/BSBW/JSB   QDMERIT
                     0225   602 ;
                     0225   603 ; INPUT PARAMETERS:
                     0225   604 ;
                     0225   605 ;       R4:             Address of CLUB
                     0225   606 ;       R9:             Address of CLUOPT block describing the tentative cluster
                     0225   607 ;       R10:            Address of vector mapping bitmap indices to CSB addresses
                     0225   608 ;
                     0225   609 ; OUTPUT PARAMETERS:
                     0225   610 ;
                     0225   611 ;       R0 contains the quorum disk's contribution to the figure of merit
                     0225   612 ;
                     0225   613 ; COMPLETION CODES:
                     0225   614 ;
                     0225   615 ;       NONE
                     0225   616 ;
                     0225   617 ; SIDE EFFECTS:
                     0225   618 ;
                     0225   619 ;       The contents of R1 and R2 are destroyed.
                     0225   620 ;
                     0225   621 ;--
                     0225   622
                     0225   623 QDMERIT:
52   00AE C4   3C   0225   624          MOVZWL  CLUB$W_QDVOTES(R4),R2   ; Accumulate minimum quorum disk votes in CM
       14 A9   9F   022A   625          PUSHAB  CLUOPT$B_CMAP(R9)       ; Push address of map to process
          1A   10   022D   626          BSBB    SCAN_MAP                ; Get call-back for each bit
     12 50     E9   022F   627          BLBC    R0,20$                  ; Branch if done
50   0C AA41   D0   0232   628          MOVL    12(R10)[R1],R0          ; CSB address
          0A   13   0237   629          BEQL    10$                     ; Branch if no CSB (quorum disk case)
  52   56 A0   B1   0239   630          CMPW    CSB$W_QDVOTES(R0),R2    ; Votes proposed for quorum disk
          04   1E   023D   631          BGEQU   10$                     ; Branch if old was lower
  52   56 A0   3C   023F   632          MOVZWL  CSB$W_QDVOTES(R0),R2    ; Update minimum
               05   0243   633 10$:     RSB                             ; Continue scanning bits
                    0244   634
50   52   08   78   0244   635 20$:     ASHL    #8,R2,R0                ; Scale votes
               05   0248   636          RSB                             ; Return, votes in R0
```

```
                     0249   638          .SBTTL  SCAN_MAP - Scan bits in a specified bitmap
                     0249   639
                     0249   640 ;++
                     0249   641 ;
                     0249   642 ; FUNCTIONAL DESCRIPTION:
                     0249   643 ;
                     0249   644 ;       This routine is called to scan all of the bits in one of the bitmaps
                     0249   645 ;       (CMAP, AMAP, RMAP) used in the optimization computation.  For every
                     0249   646 ;       bit in the map that is set, a co-routine call-back is made.
                     0249   647 ;
                     0249   648 ; CALLING SEQUENCE:
                     0249   649 ;
                     0249   650 ;       BSBB/BSBW/JSB   SCAN_MAP
                     0249   651 ;
                     0249   652 ; INPUT PARAMETERS:
                     0249   653 ;
                     0249   654 ;       4(SP):          Address of bitmap to scan
                     0249   655 ;       0(SP):          Return/co-routine address
                     0249   656 ;
                     0249   657 ; OUTPUT PARAMETERS/COMPLETION CODES:
                     0249   658 ;
                     0249   659 ;       On a co-routine callback:
                     0249   660 ;
                     0249   661 ;               R0 has the low bit set
                     0249   662 ;               R1 contains the index of the bit to process
                     0249   663 ;
                     0249   664 ;               The co-routine must return with R1 intact.
                     0249   665 ;
                     0249   666 ;       After the last bit has been processed
                     0249   667 ;
                     0249   668 ;               R0 has the low bit clear
                     0249   669 ;
                     0249   670 ; SIDE EFFECTS:
                     0249   671 ;
                     0249   672 ;       At the final return, R1 has been destroyed.
                     0249   673 ;       Any registers modified by the co-routines are changed.
                     0249   674 ;
                     0249   675 ;--
                     0249   676
                     0249   677 SCAN_MAP:
                51   D4   0249   678          CLRL    R1                              ; Initialize bit number
                     024B   679          ASSUME  CLUOPT$S_CMAP,GE,4              ; Assume at least one longword of bitmap
          50    20   D0   024B   680 10$:     MOVL    #32,R0                          ; Do as many bits as VAX can
51  04 BE   50    51   EA   024E   681 20$:     FFS     R1,R0,@4(SP),R1                 ; Look for a bit in the map
                08   13   0254   682          BEQL    30$                             ; No bits found
          50    00'  D0   0256   683          MOVL    S^#SS$_NORMAL,R0                ; Set success status
             00 BE   16   0259   684          JSB     @(SP)                           ; Do co-routine callback
                51   D6   025C   685          INCL    R1                              ; Bump over selected bit
50  000000E0 8F   51   C3   025E   686 30$:     SUBL3   R1, -                           ; Is there at least a longword left?
                     0266   687                  #<CLUOPT$S_CMAP*8>-32,R0
                E3   18   0266   688          BGEQ    10$                             ; Branch if yes
          50    20   C0   0268   689          ADDL2   #32,R0                          ; Compute number of bits remaining
                E1   14   026B   690          BGTR    20$                             ; Branch if some bits left
          6E    8E   D0   026D   691          MOVL    (SP)+,(SP)                      ; Remove map address
                50   D4   0270   692          CLRL    R0                              ; Set return status
                     05   0272   693          RSB                                     ; Return, scanning complete
                     0273   694
```

```
0273   695          .END
```

```
ADD_CMAP                        000001D6 R      02
BUG$_CNXMGRERR                  ******** X      02
CLU$GL_CLUB                     ******** X      02
CLUB$B_NODEMAP                = 000000EC
CLUB$L_FMERIT                 = 000000A8
CLUB$S_NODEMAP                = 00000020
CLUB$W_QDVOTES                = 000000AE
CLUOPT$B_AMAP                 = 00000034
CLUOPT$B_CMAP                 = 00000014
CLUOPT$B_RMAP                 = 00000054
CLUOPT$B_SUBTYPE              = 0000000B
CLUOPT$K_LENGTH               = 00000074
CLUOPT$L_ACMERIT              = 00000010
CLUOPT$L_CMERIT               = 0000000C
CLUOPT$L_PREV                 = 00000000
CLUOPT$S_AMAP                 = 00000020
CLUOPT$S_CMAP                 = 00000020
CLUOPT$S_RMAP                 = 00000020
CLUOPT$W_SIZE                 = 00000008
CNX$ALLOZMEM                    ******** X      02
CNX$OPT                         00000017 RG     02
CNX$OPT_INIT                    00000000 RG     02
CNX$SCAN_CSBS                   ******** X      02
CSB$B_NODEMAP                 = 0000008C
CSB$L_STATUS                  = 00000060
CSB$S_NODEMAP                 = 00000020
CSB$V_LOCAL                   = 00000018
CSB$V_QF_ACTIVE               = 00000009
CSB$V_QF_SAME                 = 00000003
CSB$V_SELECTED                = 00000011
CSB$W_CSID_IDX                = 0000004C
CSB$W_QDVOTES                 = 00000056
CSB$W_VOTES                   = 00000050
DYN$C_CLU_CLUOPT              = 00000006
DYN$C_CLU_CLUVEC              = 00000002
EXE$ALONONPAGED                 ******** X      02
EXE$DEANONPAGED                 ******** X      02
FKB$B_TYPE                    = 0000000A
QDMERIT                         00000225 R      02
REMOVE_AMAP                     00000209 R      02
SCAN_MAP                        00000249 R      02
SS$_NORMAL                      ******** X      02
```

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
$$$100	00000273 (627.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	36	00:00:00.06	00:00:01.85
Command processing	126	00:00:00.43	00:00:02.88
Pass 1	218	00:00:03.66	00:00:16.20
Symbol table sort	0	00:00:00.43	00:00:00.66
Pass 2	136	00:00:01.19	00:00:04.67
Symbol table output	6	00:00:00.05	00:00:00.15
Psect synopsis output	2	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	526	00:00:05.83	00:00:26.42

The working set limit was 1500 pages.
31048 bytes (61 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 429 non-local and 40 local symbols.
695 source lines were read in Pass 1, producing 14 object records in Pass 2.
14 pages of virtual memory were used to define 13 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYSLOA.OBJ]CLUSTER.MLB;1	0
_255DUA28:[SYS.OBJ]LIB.MLB;1	6
_255DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	10

502 GETS were required to define 10 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:CNXOPT/OBJ=OBJ$:CNXOPT MSRC$:CNXOPT/UPDATE=(ENH$:CNXOPT)+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