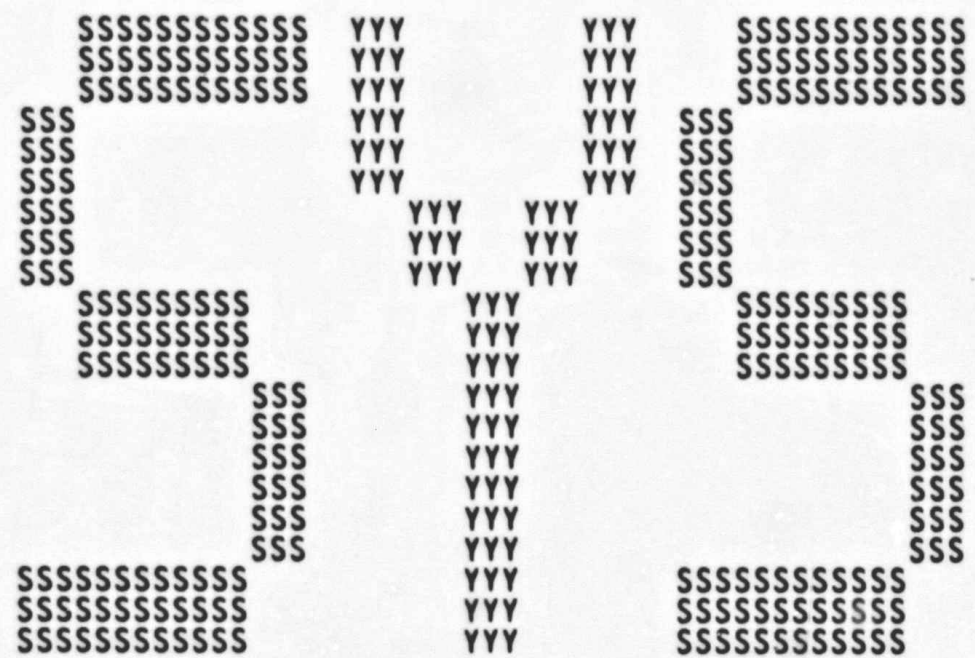




[illegible]

(2) 132 DECLARATIONS


```
0000 1      .IF      NDF,PRMSW
0000 2      .TITLE   CLUSTERVEC - Cluster Loadable Code Vectors
0000 3      .IFF
0000 4      .TITLE   CLUSTERLOA - Cluster Loadable Code
0000 5      .ENDC
0000 6      .IDENT   'V04-000'
0000 7
0000 8
0000 9 *****
0000 10
0000 11 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 12 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 13 *  ALL RIGHTS RESERVED.
0000 14 *
0000 15 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 16 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 17 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 18 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 19 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 20 *  TRANSFERRED.
0000 21 *
0000 22 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 23 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 24 *  CORPORATION.
0000 25 *
0000 26 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 27 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 28 *
0000 29 *
0000 30 *****
0000 31
0000 32
0000 33 ++
0000 34
0000 35 FACILITY:      Executive, system services and fork level code
0000 36
0000 37 ABSTRACT:
0000 38 This module contains the entry point vectors used to implement
0000 39 VMS cluster functions as well as the table used to hook up those
0000 40 vectors to the actual routines. This module is assembled in two
0000 41 different ways. If PRMSW is defined the resultant module (CLUSTERLOA)
0000 42 is linked with the actual loadable routines. If PRMSW is undefined,
0000 43 the resultant module (CLUSTERVEC) is linked with SYS.EXE.
0000 44
0000 45 ENVIRONMENT:   Kernel mode, fork level
0000 46
0000 47 --
0000 48
0000 49 AUTHOR: Steve Beckhardt,      CREATION DATE: 6-Jan-1983
0000 50
0000 51 MODIFIED BY:
0000 52
0000 53 V03-020 DWT0239      David W. Thiel      29-Aug-1984
0000 54 Add CNX$BUGCHECK_CLUSTER entry point.
0000 55
0000 56 V03-019 DWT0207      David W. Thiel      09-Apr-1984
0000 57 Add CNX$POWER_FAIL vector to be called on a
```



```
0000 58 : recovery from a power failure. Add CNX$DISK_CHANGE
0000 59 : so that SYSINIT and CSP can inform connection
0000 60 : manager when quorum disk is found.
0000 61 :
0000 62 : V03-018 SRB0117 Steve Beckhardt 18-Mar-1984
0000 63 : Added vectors to get to routines for distributed
0000 64 : deadlock detection.
0000 65 :
0000 66 : V03-017 DWT0177 David W. Thiel 27-Feb-1984
0000 67 : Remove CNX$DEALL_WARMCDRP and CNX$DEALL_MSG_BUF.
0000 68 : Add CNX$SHUTDOWN entry point. Rearrange code to
0000 69 : be more conservative about deallocating code under
0000 70 : the PC. Change the trailer PSECT name to be later
0000 71 : in the collating sequence. Add patch space.
0000 72 :
0000 73 : V03-016 ADE0001 Alan D. Eldridge 10-Jan-1983
0000 74 : Add EXE$CSP_BRDCST to vectors.
0000 75 :
0000 76 : V03-015 ADE0001 Alan D. Eldridge 10-Jan-1983
0000 77 : Add EXE$CSP_COMMAND to vectors and call to CSP$INIT in
0000 78 : initialization call.
0000 79 :
0000 80 : V03-014 RSH0051 R. Scott Hanna 11-AUG-1983
0000 81 : Add call to CNX$QUORUM_INIT
0000 82 :
0000 83 : V03-013 ROW0197 Ralph O. Weber 29-JUL-1983
0000 84 : Add LOADVEC definitions for CNX$PARTNER_RESPOND, a respond to
0000 85 : block transfer service, and CNX$CHANGE_QUORUM, a hook for a
0000 86 : future change QUORUM dynamically service.
0000 87 :
0000 88 : V03-012 ROW0185 Ralph O. Weber 24-JUN-1983
0000 89 : Add yet another ton of LOADVEC statements to define entries
0000 90 : for acknowledged message block transfer services.
0000 91 :
0000 92 : V03-011 RNG0011 Rod N. Gamache 13-Jun-1983
0000 93 : Add entry points for the distributed GETLKI system service.
0000 94 :
0000 95 : V03-010 JLV0268 Jake VanNoy 27-MAY-1983
0000 96 : Add EXE$CSP_BRKTHRU, entry point for cluster broadcasts.
0000 97 :
0000 98 : V03-009 ROW0179 Ralph O. Weber 29-APR-1983
0000 99 : Add numerous LOADVEC statements to define entries for new
0000 100 : acknowledged message services and distributed lock manager
0000 101 : support.
0000 102 :
0000 103 : V03-008 PRB0162 Paul R. Beck 14-APR-1983
0000 104 : Add vectors for EXE$ALLOC_CSD, EXE$DEALLOC_CSD, EXE$CSP_CALL.
0000 105 :
0000 106 : V03-007 JWH0212 Jeffrey W. Horn 13-Apr-1983
0000 107 : Fix mistype in JWH0207.
0000 108 :
0000 109 : V03-006 JWH0207 Jeffrey W. Horn 12-Apr-1983
0000 110 : Use SLVTAB macro to generate load-code prologue.
0000 111 :
0000 112 : V03-005 ROW0174 Ralph O. Weber 29-MAR-1983
0000 113 : Add LOADVEC definitions for CNX$ALLOC_CDRP, CNX$SEND_MSG,
0000 114 : CNX$SEND_MSG_RESP, and CNX$SEND_MSG_RSPID; all routines which
```


0000	115	:	
0000	116	:	the common journaling facility must call to perform cluster
0000	117	:	journaling.
0000	118	:	
0000	119	:	V03-004 DWT0087 David W. Thiel 23-Mar-1983
0000	120	:	Correct previous modification to return valid
0000	121	:	status after initialization.
0000	122	:	
0000	123	:	V03-003 SRB0069 Steve Beckhardt 11-Mar-1983
0000	124	:	Added support to allow SYSENQDEQ to be linked with
0000	125	:	CLUSTERLOA for debugging purposes. Removed cell
0000	126	:	LCK\$GL_RQSEQNM (it is now in SYSCOMMON).
0000	127	:	
0000	128	:	V03-002 DWT0070 David W. Thiel 28-Jan-1983
0000	129	:	Setup to initialize automatically on being loaded.
0000	130	:	Adjust psects and alignment.

```
0000 132      .SBTTL  DECLARATIONS
0000 133      :
0000 134      : INCLUDE FILES:
0000 135      :
0000 136      :
0000 137      :
0000 138      : MACROS:
0000 139      :
0000 140      :
0000 141      .IF      DF,PRMSW
0000 142      $DYNDDEF
0000 143      .IF TRUE_FALSE
0000 144      $SLVDEF
0000 145      :
0000 146      :
0000 147      : EQUATED SYMBOLS:
0000 148      :
0000 149      :
0000 150      :
0000 151      : VECTOR LIST
0000 152      :
0000 153      :
0000 154      .IF_TRUE
0000 155      :
0000 156      .PSECT   ___999,4
0000 157      :
0000 158      PATCH_BEGIN:
0000 159      .BLKB    256
0000 160      PATCH_END:
0000 161      :
0000 162      CLULOA_END:
0000 163      :
0000 164      .PSECT   $$$000,4
0000 165      :
0000 166      CLULOA_START:
0000 167      SLVTAB   END      =
0000 168      INITRTN  =
0000 169      SUBTYP   =
0000 170      FACILITY=
0000 171      :
0000 172      .IF FALSE
00000000 173      .PSECT   $$$500, LONG
0000 174      :
0000 175      .ALIGN   LONG
00000000 0000 176      CLUSGL_CLUB::
0000 177      .LONG    0
00000000 0004 178      CLUSGL_CLUSVEC::
0000 179      .LONG    0
0000 0008 180      CLUSGW_MAXINDEX::
0000 0008 181      .WORD    0
000A 000A 182      CLU_RSB:
05 000A 183      RSB
000B 184      :
000B 185      .ALIGN   LONG
000C 186      CLUSAL_LOAVEC::
000C 187      CLSSAL_LOAVEC::
000C 188      .IF TRUE_FALSE
```

; Data structure names
; Loadable vector definitions
; For inclusion with loadable code
; Octaword alignment
; Patch space
; End of loadable code
CLULOA_END, -
CLULOA_INIT, -
DYN\$C [C CLS, -
<CLUSTERS>
; For inclusion with SYS.EXE
; Address of CLUster Block
; Address of Cluster System vector
; Maximum index+1 in Cluster System vector
; Used to make unloaded entry a NOP
; New symbol to replace next


```
000C 189
000C 190 LOADVEC CLUSGL LOA ADDR,1,,CLULOA_START ; Cluster code load address
0010 191 LOADVEC LCKSSND_CVTREQ
0016 192 LOADVEC LCKSSND_LOCKREQ
001C 193 LOADVEC LCKSSND_GRANTED
0022 194 LOADVEC LCKSSND_DEQGR
0028 195 LOADVEC LCKSSND_DEQCV
002E 196 LOADVEC LCKSSND_DEQWT
0034 197 LOADVEC LCKSSND_BLKING
003A 198 LOADVEC LCKSSND_RMVDIR
0040 199 LOADVEC LCKSSND_TIMESTAMP_RQST
0046 200 LOADVEC LCKSSND_SRCHDLCK
004C 201 LOADVEC LCKSSND_DLCKFND
0052 202 LOADVEC LCKSSND_REDO_SRCH
0058 203 LOADVEC LCKSCVT_ID TO LKB
005E 204 LOADVEC CNX$ALLOC_CDRP
0064 205 LOADVEC CNX$ALLOC_CDRP_ONLY
006A 206 LOADVEC CNX$ALLOC_WARMCDRP
0070 207 LOADVEC CNX$ALLOC_WARMCDRP_CSB
0076 208 LOADVEC CNX$DEALL_MSG_BUF_CSB
007C 209 LOADVEC CNX$DEALL_WARMCDRP_CSB
0082 210 LOADVEC CNX$INIT_CDRP
0088 211 LOADVEC CNX$SEND_MNY_MSGS
008E 212 LOADVEC CNX$SEND_MSG
0094 213 LOADVEC CNX$SEND_MSG_CSB
009A 214 LOADVEC CNX$SEND_MSG_RESP
00A0 215 LOADVEC CNX$SEND_MSG_RSPID
00A6 216 LOADVEC CNX$BLOCK_XFER
00AC 217 LOADVEC CNX$BLOCK_XFER_IRP
00B2 218 LOADVEC CNX$PARTNER_INIT_CSB
00B8 219 LOADVEC CNX$BLOCK_READ
00BE 220 LOADVEC CNX$BLOCK_READ_IRP
00C4 221 LOADVEC CNX$BLOCK_WRITE
00CA 222 LOADVEC CNX$BLOCK_WRITE_IRP
00D0 223 LOADVEC CNX$PARTNER_FINISH
00D6 224 LOADVEC CNX$PARTNER_RESPOND
00DC 225 LOADVEC CNX$CHANGE_QUORUM
00E2 226 LOADVEC CNX$SHUTDOWN
00E8 227 LOADVEC CNX$POWER_FAIL,,CLU_RSB
00EE 228 LOADVEC CNX$DISK_CHANGE
00F4 229 LOADVEC CNX$BUGCHECK_CLUSTER
00FA 230 LOADVEC EXE$ALLOC_CSD
0100 231 LOADVEC EXE$DEALLOC_CSD
0106 232 LOADVEC EXE$CSP_BRDCT
010C 233 LOADVEC EXE$CSP_CALL
0112 234 LOADVEC EXE$CSP_COMMAND
0118 235 LOADVEC EXE$CSP_BRKTHRU
011E 236 LOADVEC LKISSND_STDREQ
0124 237 LOADVEC LKISSND_BLKING
012A 238 LOADVEC LKISSND_BLKBY
0130 239 LOADVEC LKISSND_LOCKS
0136 240
0136 241 .IF TRUE ; End of vector list
0136 242 .LONG -1
0136 243
0136 244 CLULOA_INIT: ; Initialization code
0136 245 BSBW CNX$INIT ; Do initialization
```



```
0136 246      BSBW      CNX$QUORUM_INIT      ; Perform quorum file initialization
0136 247      BLBC      R0,20$              ; Branch on error
0136 248      BSBW      CSP$INIT            ; Do load CSP code initialization
0136 249
0136 250      ; If the $ENQ and $DEQ system services (module SYSENQDEQ) are linked
0136 251      ; with this (for debugging purposes) then hook up the system service
0136 252      ; vectors to point these routines.
0136 253      ; NOTE: The SYSGEN parameter SYSPAGING MUST be set to 0 for this
0136 254      ; to work.
0136 255
0136 256      .WEAK      EXE$ENQ,EXE$DEQ
0136 257
0136 258      TSTL      #EXE$ENQ                  ; Is loadable $ENQ service present?
0136 259      BEQL      10$                      ; Not linked with this module
0136 260      JSB       G^INISWRITABLE          ; Make system writeable
0136 261      MOVW      B^30$,G^EXE$ENQ+2      ; Move JMP @# to start of resident
0136 262      MOVW      B^30$,G^EXE$DEQ+2      ; services
0136 263      MOVAB     W^EXE$ENQ+2,G^EXE$ENQ+4; Move addresses of loadable services
0136 264      MOVAB     W^EXF$DEQ+2,G^EXE$DEQ+4; to complete JMP @# instructions
0136 265      JSB       G^INISRDONLY           ; Make system read only again
0136 266
0136 267 10$:     MOVAB     G^EXE$GL_NONPAGED+4,R3 ; Address of non-paged pool listhead
0136 268      MOVAB     W^CLULOA_START,R0      ; Address of block to deallocate
0136 269      MOVZWL     #CLULOA_INIEND-CLULOA_START,R1 ; Length of block to deallocate
0136 270      SUBW3     R1,8(R0),W^CLULOA_INIEND+8 ; Compute new length
0136 271      BRW       DEALLOCATE            ; Deallocate this piece of memory
0136 272
0136 273 20$:     RSB                      ; Return
0136 274
0136 275 30$:     JMP       G^EXE$ENQ          ; Absolute jump instruction
0136 276
0136 277      ; Descriptor for patch space available in this module
0136 278      ; Note that this space is deallocated when the code is loaded
0136 279      ;
0136 280      CNX$PATCH::
0136 281          .WORD     PATCH_END-PATCH_BEGIN
0136 282          .WORD     0
0136 283          .LONG     PATCH_BEGIN
0136 284
0136 285          .PSECT    $$$010,4              ; Octaword alignment
0136 286
0136 287      ; When initialization is complete and the initialization code is
0136 288      ; deleted, this block is left at the head of the remaining code.
0136 289      ;
0136 290          .ALIGN     4
0136 291      CLULOA_INIEND:
0136 292          SLVTAB     END      =      CLULOA_END, -
0136 293                      SUBTYP =      DYN$C [C CLS, -
0136 294                      FACILITY=      <CLUSTERS>
0136 295
0136 296
0136 297      DEALLOCATE:
0136 298          JSB       G^EXE$DEALLOCATE      ; Free initialization code
0136 299          MOVZWL     S^#$$$_NORMAL,R0    ; Set success status
0136 300          RSB                      ; Return from initialization
0136 301
0136 302      .ENDC
```

CLUSTERVEC
V04-000

- Cluster Loadable Code Vectors L 6
DECLARATIONS

16-SEP-1984 00:04:13 VAX/VMS Macro V04-00
5-SEP-1984 03:40:33 [SYS.SRC]CLUSTER.MAR;1

Page 7
(2)

0136 303
0136 304 .END

CLUSTRVEC
Symbol table

- Cluster Loadable Code Vectors

M 6

16-SEP-1984 00:04:13
5-SEP-1984 03:40:33

VAX/VMS Macro V04-00
[SYS.SRC]CLUSTER.MAR;1

Page 8
(2)

CLSSAL_LOAVEC	0000000C	RG	02
CLUSAL_LOAVEC	0000000C	RG	02
CLUSGL_CLUB	00000000	RG	02
CLUSGL_CLUSVEC	00000004	RG	02
CLUSGL_LOA_ADDR	0000000C	RG	02
CLUSGW_MAXINDEX	00000008	RG	02
CLU_RSB	0000000A	R	02
CNX\$ALLOC_CDRP	00000005E	RG	02
CNX\$ALLOC_CDRP_ONLY	000000064	RG	02
CNX\$ALLOC_WARMCDRP	00000006A	RG	02
CNX\$ALLOC_WARMCDRP_CSB	000000070	RG	02
CNX\$BLOCK_READ	0000000B8	RG	02
CNX\$BLOCK_READ_IRP	0000000BE	RG	02
CNX\$BLOCK_WRITE	0000000C4	RG	02
CNX\$BLOCK_WRITE_IRP	0000000CA	RG	02
CNX\$BLOCK_XFER	0000000A6	RG	02
CNX\$BLOCK_XFER_IRP	0000000AC	RG	02
CNX\$BUGCHECK_CLUSTER	0000000F4	RG	02
CNX\$CHANGE_QUORUM	0000000DC	RG	02
CNX\$DEALL_MSG_BUF_CSB	000000076	RG	02
CNX\$DEALL_WARMCDRP_CSB	00000007C	RG	02
CNX\$DISK_CHANGE	0000000EE	RG	02
CNX\$INIT_CDRP	000000082	RG	02
CNX\$PARTNER_FINISH	0000000D0	RG	02
CNX\$PARTNER_INIT_CSB	0000000B2	RG	02
CNX\$PARTNER_RESPOND	0000000D6	RG	02
CNX\$POWER_FAIL	0000000E8	RG	02
CNX\$SEND_ANY_MSGS	000000088	RG	02
CNX\$SEND_MSG	00000008E	RG	02
CNX\$SEND_MSG_CSB	000000094	RG	02
CNX\$SEND_MSG_RESP	00000009A	RG	02
CNX\$SEND_MSG_RSPID	0000000A0	RG	02
CNX\$SHUTDOWN	0000000E2	RG	02
EXES\$ALLOC_CSD	0000000FA	RG	02
EXESCSP_BRDCST	000000106	RG	02
EXESCSP_BRKTHRU	000000118	RG	02
EXESCSP_CALL	00000010C	RG	02
EXESCSP_COMMAND	000000112	RG	02
EXES\$DEALLOC_CSD	000000100	RG	02
EXES\$LOAD_ERROR	*****	X	02
LCK\$CVT_ID_TO_LKB	000000058	RG	02
LCK\$SND_BLKING	000000034	RG	02
LCK\$SND_CVTREQ	000000010	RG	02
LCK\$SND_DEQCV	000000028	RG	02
LCK\$SND_DEQGR	000000022	RG	02
LCK\$SND_DEQWT	00000002E	RG	02
LCK\$SND_DLCKFND	00000004C	RG	02
LCK\$SND_GRANTED	00000001C	RG	02
LCK\$SND_LOCKREQ	000000016	RG	02
LCK\$SND_REDO_SRCH	000000052	RG	02
LCK\$SND_RMVDTR	00000003A	RG	02
LCK\$SND_SRCHDLCK	000000046	RG	02
LCK\$SND_TIMESTAMP_RQST	000000040	RG	02
LKISSND_BLKBY	00000012A	RG	02
LKISSND_BLKING	000000124	RG	02
LKISSND_LOCKS	000000130	RG	02
LKISSND_STDREQ	00000011E	RG	02

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 (0.)	00 (0.)	NOPIC USR
\$AB\$\$	00000000 (0.)	01 (1.)	NOPIC USR
\$\$\$500	00000136 (310.)	02 (2.)	NOPIC USR

CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE
CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	LONG

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.07	00:00:01.07
Command processing	130	00:00:00.54	00:00:03.79
Pass 1	159	00:00:04.38	00:00:11.68
Symbol table sort	0	00:00:00.06	00:00:00.09
Pass 2	72	00:00:01.24	00:00:03.54
Symbol table output	8	00:00:00.08	00:00:00.10
Psect synopsis output	1	00:00:00.02	00:00:00.15
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	407	00:00:06.39	00:00:20.43

The working set limit was 1200 pages.

20019 bytes (40 pages) of virtual memory were used to buffer the intermediate code.

There were 10 pages of symbol table space allocated to hold 82 non-local and 0 local symbols.

304 source lines were read in Pass 1, producing 18 object records in Pass 2.

10 pages of virtual memory were used to define 8 macros.

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

Macro library name	Macros defined
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	2
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	3
TOTALS (all libraries)	5

142 GETS were required to define 5 macros.

There were no errors, warnings or information messages.

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

0373

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY