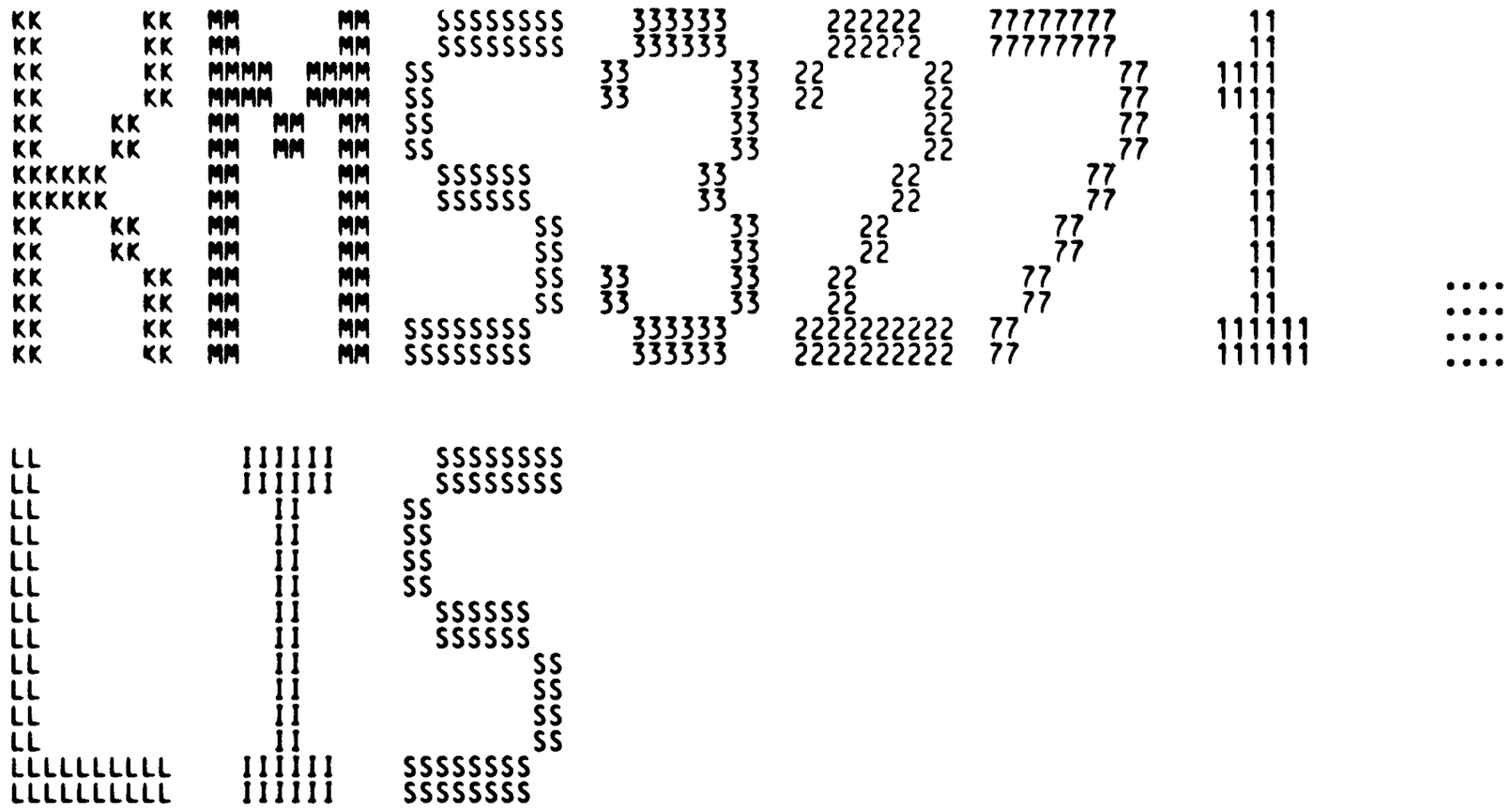
```
EEEEEEEEEEEEEEE  RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEEEEEEEEEEEEEE  RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEEEEEEEEEEEEEE  RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEEEEEEEEEEE     RRRRRRRRRRRR     FFFFFFFFFFFF
EEEEEEEEEEEE     RRRRRRRRRRRR     FFFFFFFFFFFF
EEEEEEEEEEEE     RRRRRRRRRRRR     FFFFFFFFFFFF
EEE              RRR   RRR        FFF
EEE              RRR   RRR        FFF
EEE              RRR   RRR        FFF
EEE              RRR      RRR     FFF
EEE              RRR      RRR     FFF
EEE              RRR      RRR     FFF
EEEEEEEEEEEEEEE  RRR         RRR  FFF
EEEEEEEEEEEEEEE  RRR         RRR  FFF
EEEEEEEEEEEEEEE  RRR         RRR  FFF
```

FILEID**KMS3271



```
;  1          0001  0 MODULE KMS3271_DEVICE
;  2          0002  0 (%TITLE 'KMS3271 device dependent'
;  3          0003  0 IDENT = 'V04-000') =
;  4          0004  1 Begin
;  5          0005  1
;  6          0006  1
;  7          0007  1 !
;  8          0008  1 !*****************************************************************************
;  9          0009  1 !*                                                                           *
; 10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
; 11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
; 12          0012  1 !*  ALL RIGHTS RESERVED.                                                     *
; 13          0013  1 !*                                                                           *
; 14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
; 15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
; 17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
; 18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
; 19          0019  1 !*  TRANSFERRED.                                                             *
; 20          0020  1 !*                                                                           *
; 21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
; 22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
; 23          0023  1 !*  CORPORATION.                                                             *
; 24          0024  1 !*                                                                           *
; 25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
; 26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
; 27          0027  1 !*                                                                           *
; 28          0028  1 !*                                                                           *
; 29          0029  1 !*****************************************************************************
; 30          0030  1
; 31          0031  1 !++
; 32          0032  1 ! FACILITY:  ERF, Error Log Report Generator
; 33          0033  1 !
; 34          0034  1 ! ABSTRACT:
; 35          0035  1 !
; 36          0036  1 !       This module contains the routines that handle the bit to text
; 37          0037  1 !       translation (device dependent) for the KMS3271.
; 38          0038  1 !
; 39          0039  1 ! ENVIRONMENT:
; 40          0040  1 !
; 41          0041  1 !       VAX/VMS operating system, user mode.
; 42          0042  1 !
; 43          0043  1 ! AUTHOR:  Sharon Reynolds,     CREATION DATE:  20-Feb-1984
; 44          0044  1 !
; 45          0045  1 ! Modified by:
; 46          0046  1 !
; 47          0047  1 !       V04-002 SAR0282         Sharon A. Reynolds              2-Jul-1984
; 48          0048  1 !               - Removed the output_lines routine and moved it to the
; 49          0049  1 !               shareable image.
; 50          0050  1 !               - Removed the build_fao_string macro and moved to
; 51          0051  1 !               erfdef.req.
; 52          0052  1 !
; 53          0053  1 !       V04-001 EAD0148         Elliott A. Drayton              13-Apr-1984
; 54          0054  1 !               Changed subroutine names which clashed with UCB definitions.
; 55          0055  1 !
; 56          0056  1 !--
; 57          0057  1 REQUIRE 'SRC$:RECSELDEF';
```

```
;    58          0188  1 REQUIRE 'SRC$:ERFDEF' ;
;    59          0474  1
;    60          0475  1 FORWARD ROUTINE
;    61          0476  1     Kms3271:            NOVALUE,
;    62          0477  1     Output_registers:   NOVALUE ;
;    63          0478  1
;    64          0479  1 EXTERNAL
;    65          0480  1     Emb:                $BBLOCK PSECT (EMB),
;    66          0481  1     Syecom:             $BBLOCK PSECT (SYECOM);
;    67          0482  1
;    68          0483  1 EXTERNAL ROUTINE
;    69          0484  1     Frctof,
;    70          0485  1     Dhead1,
;    71          0486  1     Orb$l_owner_rtn,
;    72          0487  1     Output_lines,
;    73          0488  1     Translate_bits,
;    74          0489  1     Uba_mapping,
;    75          0490  1     Ucb$l_char,
;    76          0491  1     Ucb$w_sts_rtn,
;    77          0492  1     Ucb$l_opcnt_rtn,
;    78          0493  1     Ucb$w_errcnt_rtn ;
;    79          0494  1
;    80          0495  1 OWN
;    81          0496  1     Lstlun:             LONG,
;    82          0497  1     Arglist:            Vector [10,long],
;    83          0498  1     Bts_sts_codes:      Vector [28,byte]
;    84          0499  1                         Initial (byte(%x'2',%x'4',%x'6',%x'8',%x'9',%x'a',
;    85          0500  1                         %x'c',%x'e',%x'10',%x'27',%x'28',%x'2d',%x'2e',
;    86          0501  1                         %x'2f',%x'30',%x'31',%x'32',%x'33',%x'34',%x'35',
;    87          0502  1                         %x'36',%x'37',%x'38',%x'39',%x'3a',%x'3d',%x'3e',
;    88          0503  1                         %x'3f')),
;    89          0504  1     Cmd_sts_codes:      Vector [19,byte]
;    90          0505  1                         Initial (byte(%x'9',%x'28',%x'2d',%x'2e',
;    91          0506  1                         %x'2f',%x'30',%x'31',%x'32',%x'33',%x'34',%x'35',
;    92          0507  1                         %x'36',%x'37',%x'38',%x'39',%x'3a',%x'3d',%x'3e',
;    93          0508  1                         %x'3f')),
;    94          0509  1     Csr0_mask:          Initial (%x'E791'),
;    95          0510  1     Csr1_mask:          Initial (%x'90'),
;    96          0511  1     Dev_id_desc:        Initial ( $DESCRIPTOR('UBA KMS3271') ),
;    97          0512  1     Fao_string,
;    98          0513  1     Fao_string1:        Initial (%ASCID '!/!7< !>!AC!#< !>!XW'),
;    99          0514  1     Fao_strings:        Vector [13,long]
;   100          0515  1                         Initial (long
;   101          0516  1                          (%ASCID '!39< !>LINE NUMBER = !ZB.'),
;   102          0517  1                          (%ASCID '!39< !>MEMORY EXTENSION BITS = !XB(X)!/!39< !>LINE NUMBER = !ZB.!/!39< !>L
;   103          0518  1                          (%ASCID '!39< !>LINE NUMBER = !ZB.!/!39< !>LUN = !ZB.'),
;   104          0519  1                          (%ASCID '!39< !>!AS'),
;   105          0520  1                          (%ASCID '!39< !>!AS!XL(X)'),
;   106          0521  1                          (%ASCID '!39< !>MEMORY EXTENSION BITS = !XB(X)!/!39< !>!AS'),
;   107          0522  1                          (%ASCID '!39< !>BYTE COUNT = !ZB.!/!39< !>MEMORY EXTENSION BITS = !XB(X)'),
;   108          0523  1                          (%ASCID '!39< !>BYTE COUNT = !ZB.!/!39< !>!AS'),
;   109          0524  1                          (%ASCID '!39< !>MEMORY EXTENSION BITS = !XB(X)'),
;   110          0525  1                          (%ASCID '!39< !>STATUS BYTE 0!XB(X)!/!39< !>STATUS BYTE 1!XB(X)'),
;   111          0526  1                          (%ASCID '!39< !>MEMORY EXTENSION BITS = !XB(X)!/!39< !>LINE NUMBER = !ZB.'),
;   112          0527  1                          (%ASCID 'COMMAND COMPLETED SUCCESSFULLY:!/!39< !>RECEIVED !AC BLOCK OF A!/!39< !>!A
;   113          0528  1                          (%ASCID 'COMMAND COMPLETED SUCCESSFULLY:!/!39< !>RECEIVED !AC BUFFER!/!39< !>!AC')
;   114          0529  1     Input_header:       Initial ( $DESCRIPTOR('REGISTER INPUT TO KMS') ),
```

```
;   115          0530  1     Mcu_string:         Initial (%ASCID 'MODEM CNTRL UNIT ADDRS = '),
;   116          0531  1     Mem_string:         Initial (%ASCID 'MEMORY BUFFER ADDRS = '),
;   117          0532  1     Msg:                Vector [4,long]
;   118          0533  1                         Initial ((long
;   119          0534  1                                 (CSTRING ('FIRST'),
;   120          0535  1                                  CSTRING ('LAST'),
;   121          0536  1                                  CSTRING ('INTERMEDIATE'),
;   122          0537  1                                  CSTRING ('ONLY'))),
;   123          0538  1     Out_arglist:        REF VECTOR [31,long]
;   124          0539  1                         Initial (0),
;   125          0540  1     Output_header:      Initial ( $DESCRIPTOR('REGISTER OUTPUT FROM KMS') ),
;   126          0541  1     Status,
;   127          0542  1     Unknown_desc:       Initial ($DESCRIPTOR ('UNKNOWN STATUS CODE = !XB(X)')) ;
;   128          0543  1
;   129          0544  1 !
;   130          0545  1 ! Define some macros.
;   131          0546  1 !
;   132          0547  1 ! Define some offsets for data in the registers.
;   133          0548  1 !
;   134          0549  1 MACRO
;   135          0550  1     Byte_cnt    = 0,0,10,0%,
;   136          0551  1     Cmd         = 0,0,4,0%,
;   137          0552  1     Cmd_sts     = 0,10,6,0%,
;   138          0553  1     Inp_mem_ext = 0,10,2,0%,
;   139          0554  1     Line_num    = 0,8,2,0%,
;   140          0555  1     Lun         = 0,11,5,0%,
;   141          0556  1     Mem_addrs   = 0,0,16,0%,
;   142          0557  1     Mem_ext     = 0,5,2,0%,
;   143          0558  1     Out_mem_ext = 0,2,2,0%,
;   144          0559  1     Sts_byte_0  = 0,0,8,0%,
;   145          0560  1     Sts_byte_1  = 0,8,8,0% ;
;   146          0561  1
;   147          0562  1 !
;   148          0563  1 ! Store the text necessary to translate the error log packet for this device.
;   149          0564  1 !
;   150        P 0565  1 STORE_STRINGS ( kmscmds,
;   151        P 0566  1                 'MODEM INITIALIZATION',
;   152        P 0567  1                 'LINE INITIALIZATION',
;   153        P 0568  1                 'LINE DEINITIALIZATION',
;   154        P 0569  1                 'ADDRESS DESCRIPTOR CONNECT',
;   155        P 0570  1                 'ADDRESS DESCRIPTOR DISCONNECT',
;   156        P 0571  1                 'LINE TRANSMIT BUFFER',
;   157        P 0572  1                 'LINE RECEIVE BUFFER',
;   158        P 0573  1                 'LINE TRANSMIT ABORT',
;   159        P 0574  1                 'LINE RECEIVE ABORT',
;   160        P 0575  1                 'LINE STATISTICS REQUEST',
;   161        P 0576  1                 'TRANSMIT READ COMMAND',
;   162          0577  1                 'TRANSMIT STATUS' ) ;
;   163          0578  1
;   164        P 0579  1 STORE_STRINGS ( BTSTATUS,
;   165        P 0580  1                 'UNIBUS CONTROLLER HAS STOPPED !/!39< !>ACCEPTING COMMANDS',
;   166        P 0581  1                 'COMMAND ISSUED FOR THE LINE !/!39< !>NEVER COMPLETED',
;   167        P 0582  1                 'WRITE FAILED TO COMPLETE IN THE !/!39< !>MAX TIME',
;   168        P 0583  1                 'LOAD MICROCODE FAILED',
;   169        P 0584  1                 'RECEIVED DEVICE STATUS MESSAGE',
;   170        P 0585  1                 'MICROCODE FAILED TO START',
;   171        P 0586  1                 'UNEXPECTED AND INCONSISTENT EVENT!/!39< !>CAUSED PORT DRIVER T  SHUT LINE !/!39< !>DOWN',
```

```
;  172          P 0587  1                  'NOT ENOUGH MAP REGISTERS !/!39< !>AVAILABLE FOR A WRITE',
;  173          P 0588  1                  'S/S BYTES RECEIVED FROM CLASS !/!39< !>INSTEAD OF DATA',
;  174          P 0589  1                  'STATUS RECEIVED FOR A LUN THAT !/!39< !>IS ALSO THE FIRST LUN QUEUED TO !/!39< !>TRANSMIT (
;  175          P 0590  1                  'IBM HOST HAS TIMED OUT !/!39< !>(SLAVE MODE ONLY)',
;  176          P 0591  1                  'REMOTE STATION COULD NOT BE !/!39< !>ADDRESSED',
;  177          P 0592  1                  'REMOTE STATION ABORTED A BUFFER !/!39< !>(SLAVE)',
;  178          P 0593  1                  'INVALID PARAMETERS',
;  179          P 0594  1                  'NOT ENOUGH RECEIVE BUFFERS !/!39< !>PRESENT',
;  180          P 0595  1                  'CONTROLLER FAILED TO ACCEPT !/!39< !>A COMMAND',
;  181          P 0596  1                  'MODEM QUEUE OVERFLOW',
;  182          P 0597  1                  'CSR QUEUE OVERFLOW',
;  183          P 0598  1                  'MAX MESSAGE SIZE EXCEEDED',
;  184          P 0599  1                  'MODEM FOR THIS PORT FAILED',
;  185          P 0600  1                  'MODEM CONTROL UNIT HAS FAILED !/!39< !>ON THIS PORT'
;  186          P 0601  1                  'MODEM CONTROL UNIT DOES NOT !/!39< !>RESPOND',
;  187          P 0602  1                  'MAX NUMBER OF RETRIES WAS !/!39< !>EXCEEDED',
;  188          P 0603  1                  'PORT DRIVER AND FIRMWARE ARE !/!39< !>OUT OF SYNC',
;  189          P 0604  1                  'READ BUFFER OVERFLOWED',
;  190          P 0605  1                  'COMMAND REJECT: BUFFERS PENDING',
;  191          P 0606  1                  'TOO MANY OPERATIONS OUTSTANDING',
;  192            0607  1                  'NON-EXISTENT MEMORY ERROR') ;
;  193            0608  1
;  194          P 0609  1  STORE_STRINGS ( CMDSTS,
;  195          P 0610  1                  'DEVICE STATUS RETURNED IN SEL 4 !/!39< !>(2 BYTES)',
;  196          P 0611  1                  'HOST TIMED OUT',
;  197          P 0612  1                  'BAD LUN: INVALID LUN SPECIFIED',
;  198          P 0613  1                  'ABORTED TEXT BUFFER: BUFFER !/!39< !>TERMINATED WITH ''ENQ''',
;  199          P 0614  1                  'INVALID PARAMETERS',
;  200          P 0615  1                  'NOT ENOUGH RECEIVE BUFFERS: !/!39< !>MESSAGE BEING RECEIVED FOR WHICH !/!39< !>THERE WERE N
;  201          P 0616  1                  'USER SPECIFIED AN INVALID !/!39< !>COMMAND: LINE NOT IN SLAVE MODE',
;  202          P 0617  1                  'MODEM QUEUE OVERFLOW: COMMAND !/!39< !>ABORTED DUE TO SYSTEM BACKLOG',
;  203          P 0618  1                  'CSR QUEUE OVERFLOW (32 ENTRIES): !/'39< !>INTERNAL FEP DATA STRUCTURE !/!39< !>OVERFLOW',
;  204          P 0619  1                  'N1 ERROR (EMPTY BUFFER OR ODD !/!39< !>ADDRESS COUNT EXCEEDED)',
;  205          P 0620  1                  'MODEM DOWN: PERIODIC CHECK OF !/!39< !>MCU FOUND CTS, DSR, OR CD CLEARED',
;  206          P 0621  1                  'MODEM CONTROL UNIT DOWN: !/!39< !>PERIODIC CHECK OF MCU FOUND RTS !/!39< !>OR DTR CLEARED',
;  207          P 0622  1                  'NON-EXISTENT MODEM CONTROL UNIT',
;  208          P 0623  1                  'NO RESPONSE AFTER N2 RETRIES',
;  209          P 0624  1                  'BUFFER OVERFLOW: BUFFER RECEIVED !/!39< !>WAS LARGER THAN THE RECEIVE !/!39< !>BUFFER ALLOC
;  210          P 0625  1                  'PENDING BUFFER: FEP REJECTED !/!39< !>COMMAND BECAUSE TRANSMIT OR !/!39< !>RECEIVE BUFFERS
;  211          P 0626  1                  'TOO MANY BUFFERS: THIS BUFFER !/!39< !>WOULD CAUSE FEP TO HAVE 9 BUFFERS',
;  212            0627  1                  'NON-EXISTENT MEMORY WHILE!/!39< !>ACCESSING !AC BUFFER') ;
;  213            0628  1
;  214            0629  1
;  215          P 0630  1  STORE_STRINGS ( LINE_STATES,
;  216          P 0631  1                  'OFF',
;  217          P 0632  1                  'RESOURCES ALLOCATED, NO REQUESTS !/!39< !>ISSUED TO KMS',
;  218          P 0633  1                  'TRANSITIONING UP/ATTEMPTING KMS !/!39< !>''INIT'' OR DOWN/ATTEMPTING !/!39< !>KMS ''DEINIT''',
;  219          P 0634  1                  'TRANSITIONING UP/ATTEMPTING !/!39< !>''CONNECT'' OR !/!39< !>DOWN/ATTEMPTING !/!39< !>''DISCON
;  220          P 0635  1                  'ACTIVE: MODEM OR MODEM CONTROL !/!39< !>FAILURE',
;  221          P 0636  1                  'SHUTTING DOWN: ''TRANSMIT ABORT''!/!39< !> ISSUED',
;  222          P 0637  1                  'SHUTTING DOWN: ''RECEIVE ABORT'' !/!39< !>ISSUED',
;  223            0638  1                  'ACTIVE') ;
;  224            0639  1
;  225          P 0640  1  STORE_STRINGS ( CSR0,
;  226          P 0641  1                  'INTERRUPT ENABLE IN',
;  227          P 0642  1                  'INTERRUPT ENABLE OUT',
;  228          P 0643  1                  'REQUEST IN',
```

```
;  229          P 0644  1                    'STEP UP',
;  230          P 0645  1                    'RAM IN',
;  231          P 0646  1                    'RAM OUT',
;  232          P 0647  1                    'CRAM WRITE',
;  233          P 0648  1                    'MASTER CLEAR',
;  234            0649  1                    'RUN') ;
;  235            0650  1
;  236          P 0651  1 STORE_STRINGS ( CSR1,
;  237          P 0652  1                    'READY IN',
;  238            0653  1                    'READY OUT') ;
;  239            0654  1
```

```
;  241          0655  1 GLOBAL Routine KMS3271 : NOVALUE =
;  242          0656  2 Begin
;  243          0657  2
;  244          0658  2 !++
;  245          0659  2 !
;  246          0660  2 ! Functional Description:
;  247          0661  2 !
;  248          0662  2 !    This routine contains the algorithms for translating the
;  249          0663  2 !    contents of a KMS3271 error log packet.
;  250          0664  2 !
;  251          0665  2 ! Calling Sequence:
;  252          0666  2 !
;  253          0667  2 !       KMS3271 () ;
;  254          0668  2 !
;  255          0669  2 ! Input Parameters:
;  256          0670  2 !
;  257          0671  2 !       None
;  258          0672  2 !
;  259          0673  2 ! Output Parameters:
;  260          0674  2 !
;  261          0675  2 !       None
;  262          0676  2 !
;  263          0677  2 !--
;  264          0678  2
;  265          0679  2 Bind regsav = (emb[emb$l_dv_regsav]+4) : vector [,long] ;
;  266          0680  2
;  267          0681  2 Bind btstatus = regsav[0] : byte ;
;  268          0682  2 Bind line_state = regsav[1] : byte ;
;  269          0683  2 Bind map_number = regsav[10] ;
;  270          0684  2 Bind map_contents = regsav[11] ;
;  271          0685  2 Bind final_map_contents = regsav[12] ;
;  272          0686  2
;  273          0687  2 LOCAL
;  274          0688  2     Buffer:     Vector [512,byte],
;  275          0689  2     Desc:       $BBLOCK [8]
;  276          0690  2                 Preset ( [dsc$b_dtype] = dsc$k_dtype_t,
;  277          0691  2                          [dsc$b_class] = dsc$k_class_s),
;  278          0692  2     Flag,
;  279          0693  2     Index,
;  280          0694  2     Index2 ;
;  281          0695  2
;  282          0696  2 !
;  283          0697  2 ! Initialize the buffer descriptor and lstlun.
;  284          0698  2 !
;  285          0699  2 Desc[dsc$w_length] = %ALLOCATION(buffer) ;
;  286          0700  2 Desc[dsc$a_pointer] = buffer ;
;  287          0701  2 Lstlun = .syecom[sye$l_lstlun];
;  288          0702  2
;  289          0703  2 !
;  290          0704  2 ! Force top of form and output the entry header.
;  291          0705  2 !
;  292          0706  2 FRCTOF (lstlun) ;
;  293          0707  2 DHEAD1 (lstlun,.dev_id_desc) ;
;  294          0708  2
;  295          0709  2 !
;  296          0710  2 ! Translate and output the driver status.
;  297          0711  2 !
```

```
;  298          0712  2 ! Arglist contains: the register mnemonic, the starting position of the
;  299          0713  2 ! register contents field, the register contents, and the address of the
;  300          0714  2 ! descriptor for the driver status text.
;  301          0715  2 !
;  302          0716  2 Arglist[0] = CSTRING ('DRIVER STATUS') ;
;  303          0717  2 Arglist[1] = (16 - .(.arglist[0])<0,8>)+4 ;
;  304          0718  2 Arglist[2] = .btstatus ;
;  305          0719  2 Arglist[3] = desc ;
;  306          0720  2
;  307          0721  2 !
;  308          0722  2 ! Get the driver status text.
;  309          0723  2 !
;  310          0724  2 ! Determine if the driver status code is in the bts_sts_codes table.
;  311          0725  2 !
;  312          0726  2 Flag = 0 ;
;  313          0727  2 Incr index from 0 to 27 do
;  314          0728  3     Begin
;  315          0729  3     If .btstatus EQL .bts_sts_codes[.index]
;  316          0730  3     Then
;  317          0731  3         !
;  318          0732  3         ! Yes, save the address of the descriptor, indicate that the
;  319          0733  3         ! text was found and exit the increment loop.
;  320          0734  3         !
;  321          0735  4         Begin
;  322          0736  4         Arglist[4] = btstatus_desc_tbl[.index,0,0,0,0] ;
;  323          0737  4         Flag = 1 ;
;  324          0738  4         Exitloop ;
;  325          0739  3         End ;
;  326          0740  2     End ;
;  327          0741  2
;  328          0742  2 !
;  329          0743  2 ! Determine if the text was found.
;  330          0744  2 !
;  331          0745  2 If NOT .flag
;  332          0746  2 Then
;  333          0747  2     !
;  334          0748  2     ! Determine if the driver status code was one of the successful completion
;  335          0749  2     ! codes and build the message.
;  336          0750  2     !
;  337          0751  3     Begin
;  338          0752  3     Arglist[6] = CSTRING ('DATA') ,
;  339          0753  3     Index2 = -1 ;
;  340          0754  3
;  341          0755  3     Case .btstatus from 1 to %x'17' of
;  342          0756  3         Set
;  343          0757  3         [1]: Index2 = 2 ;
;  344          0758  3         [3]: Index2 = 0 ;
;  345          0759  3         [5]: Index2 = 1 ;
;  346          0760  3         [7]: Index2 = 3 ;
;  347          0761  3         [%x'11']:
;  348          0762  4             Begin
;  349          0763  4             Index2 = 2 ;
;  350          0764  4             Arglist[6] = CSTRING ('TEST') ;
;  351          0765  3             End ;
;  352          0766  3
;  353          0767  3         [%x'13']:
;  354          0768  4             Begin
```

```
;  355     0769  4              Index2 = 0 ;
;  356     0770  4              Arglist[6] = CSTRING ('TEST') ;
;  357     0771  3              End ;
;  358     0772  3
;  359     0773  3          [%x'15']:
;  360     0774  4              Begin
;  361     0775  4              Index2 = 1 ;
;  362     0776  4              Arglist[6] = CSTRING ('TEST') ;
;  363     0777  3              End ;
;  364     0778  3
;  365     0779  3          [%x'17']:
;  366     0780  4              Begin
;  367     0781  4              Index2 = 3 ;
;  368     0782  4              Arglist[6] = CSTRING ('TEST') ;
;  369     0783  3              End ;
;  370     0784  3
;  371     0785  3          [INRANGE]:
;  372     0786  3              Index2 = -1 ;
;  373     0787  3          Tes ;
;  374     0788  3
;  375     0789  3      !
;  376     0790  3      ! Determine if driver status was one of the completion msgs.
;  377     0791  3      !
;  378     0792  3      If .index2 NEQ -1
;  379     0793  3      Then
;  380     0794  3          !
;  381     0795  3          ! Yes, it was a completion msg, get the fao control string and
;  382     0796  3          ! address of the descriptor.
;  383     0797  3          !
;  384     0798  4          Begin
;  385     0799  4          Arglist[4] = .fao_strings[11] ;
;  386     0800  4          Arglist[5] = .msg[.index2] ;
;  387     0801  4          End
;  388     0802  3      Else
;  389     0803  3          !
;  390     0804  3          ! No, get the address of the unknown msg descriptor and the driver
;  391     0805  3          ! status code.
;  392     0806  3          !
;  393     0807  4          Begin
;  394     0808  4          Arglist[4] = .unknown_desc ;
;  395     0809  4          Arglist[5] = .btstatus ;
;  396     0810  3          End ;
;  397     0811  2      End ;
;  398     0812  2
;  399     0813  2 !
;  400     0814  2 ! Format the string and output the driver status information.
;  401     0815  2 !
;  402     0816  2 FORMAT_STRING (arglist[4],desc,arglist[5]) ;
;  403     0817  2
;  404     0818  2 OUTPUT_LINES ( .fao_string1,arglist[0],
;  405     0819  2                .fao_strings[3],arglist[3]) ;
;  406     0820  2
;  407     0821  2 !
;  408     0822  2 ! Translate and output the line state.
;  409     0823  2 !
;  410     0824  2 ! Arglist contains: the register mnemonic, the starting position of the
;  411     0825  2 ! register contents field, the register contents, and the address of the
```

```
;  412          0826  2 ! descriptor for the line state text.
;  413          0827  2 !
;  414          0828  2 Arglist[0] = CSTRING ('LINE STATE') ;
;  415          0829  2 Arglist[1] = (16 - .(.arglist[0])<0,8>)+4 ;
;  416          0830  2 Arglist[2] = .line_state ;
;  417          0831  2 Arglist[3] = line_states_desc_tbl[.line_state,0,0,0,0] ;
;  418          0832  2
;  419          0833  2 !
;  420          0834  2 ! Determine if the line state text contains multiple lines.
;  421          0835  2 !
;  422          0836  2 If .line_state  NEQ 0 OR
;  423          0837  2     .line_state NEQ 7
;  424          0838  2 Then
;  425          0839  2     !
;  426          0840  2     ! Yes, format the string.
;  427          0841  2     !
;  428          0842  3     Begin
;  429          0843  3     Desc[dsc$w_length] = %ALLOCATION(buffer) ;
;  430          0844  3     FORMAT_STRING (arglist[3],desc,%REF(0)) ;
;  431          0845  3     Arglist[3] = desc ;
;  432          0846  2     End ;
;  433          0847  2
;  434          0848  2 !
;  435          0849  2 ! Output the line state information.
;  436          0850  2 !
;  437          0851  2 OUTPUT_LINES ( .fao_string1,arglist[0],
;  438          0852  2                .fao_strings[3],arglist[3] ) ;
;  439          0853  2
;  440          0854  2 !
;  441          0855  2 ! Translate and output the register input to the KMS.
;  442          0856  2 !
;  443          0857  2 OUTPUT_REGISTERS (0) ;
;  444          0858  2
;  445          0859  2 !
;  446          0860  2 ! Translate and output the register output from the KMS.
;  447          0861  2 !
;  448          0862  2 OUTPUT_REGISTERS (1) ;
;  449          0863  2
;  450          0864  2 !
;  451          0865  2 ! Determine if there is mapping information that needs to be
;  452          0866  2 ! translated and output.
;  453          0867  2 !
;  454          0868  2 If .btstatus EQL %x'3F' OR       ! non-existent memory error
;  455          0869  2     .btstatus EQL %x'E'         ! not enough map registers error
;  456          0870  2 Then
;  457          0871  2     !
;  458          0872  2     ! Yes, call the appropriate routines to translate and output it.
;  459          0873  2     !
;  460          0874  3     Begin
;  461          0875  3     UBA_MAPPING (lstlun,map_number,map_contents) ;
;  462          0876  3     UBA_MAPPING (lstlun,%REF(-1),final_map_contents) ;
;  463          0877  2     End ;
;  464          0878  2
;  465          0879  2 !
;  466          0880  2 ! Call the appropriate routines to translate and output
;  467          0881  2 ! the unit control block information.
;  468          0882  2 !
```

```
;  469          0883  2 ORB$L_OWNER_RTN (lstlun,emb[emb$l_dv_ownuic]) ;
;  470          0884  2 UCB$L_CHAR_RTN (lstlun,emb[emb$l_dv_char]) ;
;  471          0885  2 UCB$W_STS_RTN (lstlun,emb[emb$w_dv_sts]) ;
;  472          0886  2 UCB$L_OPCNT_RTN (lstlun,emb[emb$l_dv_opcnt]) ;
;  473          0887  2 UCB$W_ERRCNT_RTN (lstlun,emb[emb$w_dv_errcnt]) ;
;  474          0888  2
;  475          0889  2 !
;  476          0890  2 !UCB$B_SLAVE is currently not used but may be in the future.
;  477          0891  2 !
;  478          0892  2
;  479          0893  1 End ;   ! Routine
```

```
                                                                .TITLE  KMS3271_DEVICE KMS3271 device dependent
                                                                .IDENT  \V04-000\

                                                                .PSECT  $PLIT,NOWRT,NOEXE,  PIC,2

                31  37  32  33  53  4D  4B  20  41  42  55  00000 P.AAB:  .ASCII  \UBA KMS3271\                          ;
                                                        0000B           .BLKB   1
                                                    0000000B    0000C P.AAA:  .LONG   11                                ;
                                                    00000000'   00010         .ADDRESS P.AAB                        ;
20  3C  23  21  43  41  21  3E  21  20  3C  37  21  2F  21      00014 P.AAD:  .ASCII  \!/!7< !>!AC!#< !>!XW\              ;
                                        57  58  21  3E  21      00023                                                 ;
                                                    010E0014    00028 P.AAC:  .LONG   17694740                          ;
                                                    00000000'   0002C         .ADDRESS P.AAD                        ;
4D  55  4E  20  45  4E  49  4C  3E  21  20  3C  39  33  21      00030 P.AAF:  .ASCII  \!39< !>LINE NUMBER = !ZB.\<0><0><0>  ;
        00  00  00  2E  42  5A  21  20  3D  20  52  45  42      0003F                                                 ;
                                                    010E0019    0004C P.AAE:  .LONG   17694745                          ;
                                                    00000000'   00050         .ADDRESS P.AAF                        ;
45  20  59  52  4F  4D  45  4D  3E  21  20  3C  39  33  21      00054 P.AAH:  .ASCII  \!39< !>MEMORY EXTENSION BITS = !XB(X)!/!\ ;
3D  20  53  54  49  42  20  4E  4F  49  53  4E  45  54  58      00063                                                 ;
                    21  2F  21  29  58  28  42  58  21  20      00072                                                 ;
42  4D  55  4E  20  45  4E  49  4C  3E  21  20  3C  39  33      0007C         .ASCII  \39< !>LINE NUMBER = !ZB.!/!39< !>LUN = !\ ;
3C  39  33  21  2F  21  2E  42  5A  21  20  3D  20  52  45      0008B                                                 ;
                    21  20  3D  20  4E  55  4C  3E  21  20      0009A                                                 ;
                                        00  2E  42  5A          000A4         .ASCII  \ZB.\<0>                        ;
                                                    010E0053    000A8 P.AAG:  .LONG   17694803                          ;
                                                    00000000'   000AC         .ADDRESS P.AAH                        ;
4D  55  4E  20  45  4E  49  4C  3E  21  20  3C  39  33  21      000B0 P.AAJ:  .ASCII  \!39< !>LINE NUMBER = !ZB.!/!39< !>LUN = \ ;
39  33  21  2F  21  2E  42  5A  21  20  3D  20  52  45  42      000BF                                                 ;
                    20  3D  20  4E  55  4C  3E  21  20  3C      000CE                                                 ;
                                        2E  42  5A  21          000D8         .ASCII  \!ZB.\                          ;
                                                    010E002C    000DC P.AAI:  .LONG   17694764                          ;
                                                    00000000'   000E0         .ADDRESS P.AAJ                        ;
                00  00  53  41  21  3E  21  20  3C  39  33  21  000E4 P.AAL:  .ASCII  \!39< !>!AS\<0><0>               ;
                                                    010E000A    000F0 P.AAK:  .LONG   17694730                          ;
                                                    00000000'   000F4         .ADDRESS P.AAL                        ;
58  28  4C  58  21  53  41  21  3E  21  20  3C  39  33  21      000F8 P.AAN:  .ASCII  \!39< !>!AS!XL(X)\                ;
                                                        29      00107                                                 ;
                                                    010E0010    00108 P.AAM:  .LONG   17694736                          ;
                                                    00000000'   0010C         .ADDRESS P.AAN                        ;
45  20  59  52  4F  4D  45  4D  3E  21  20  3C  39  33  21      00110 P.AAP:  .ASCII  \!39< !>MEMORY EXTENSION BITS = !XB(X)!/!\ ;
3D  20  53  54  49  42  20  4E  4F  49  53  4E  45  54  58      0011F                                                 ;
                    21  2F  21  29  58  28  42  58  21  20      0012E                                                 ;
                00  00  00  53  41  21  3E  21  20  3C  39  33  00138         .ASCII  \39< !>!AS\<0><0><0>            ;
                                                    010E0031    00144 P.AAO:  .LONG   17694769                          ;
```

```
                                                00000000'  00148          .ADDRESS P.AAP
55 4F 43 20 45 54 59 42 3E 21 20 3C 39 33 21    00014C P.AAR:  .ASCII  \!39< !>BYTE COUNT = !ZB.!/!39< !>MEMORY \
3C 39 33 21 2F 21 2E 42 5A 21 20 3D 20 54 4E    00015B
                  20 59 52 4F 4D 45 4D 3E 21 20 00016A
20 53 54 49 42 20 4E 4F 49 53 4E 45 54 58 45    00174          .ASCII  \EXTENSION BITS = !XB(X)\<0>
                  00 29 58 28 42 58 21 20 3D    00183
                                             010E003F  0018C P.AAQ:  .LONG   17694783
                                             00000000' 00190          .ADDRESS P.AAR
55 4F 43 20 45 54 59 42 3E 21 20 3C 39 33 21    00194 P.AAT:  .ASCII  \!39< !>BYTE COUNT = !ZB.!/!39< !>!AS\
3C 39 33 21 2F 21 2E 42 5A 21 20 3D 20 54 4E    001A3
                              53 41 21 3E 21 20 001B2
                                             010E0024  001B8 P.AAS:  .LONG   17694756
                                             00000000' 001BC          .ADDRESS P.AAT
45 20 59 52 4F 4D 45 4D 3E 21 20 3C 39 33 21    001C0 P.AAV:  .ASCII  \!39< !>MEMORY EXTENSION BITS = !XB(X)\<0>
3D 20 53 54 49 42 20 4E 4F 49 53 4E 45 54 58    001CF
                  00 29 58 28 42 58 21 20 00    001DE
                                 00 00          001E6          .ASCII  <0><0>
                                             010E0025  001E8 P.AAU:  .LONG   17694757
                                             00000000' 001EC          .ADDRESS P.AAV
42 20 53 55 54 41 54 53 3E 21 20 3C 39 33 21    001F0 P.AAX:  .ASCII  \!39< !>STATUS BYTE 0!XB(X)!/!39< !>STATU\
33 21 2F 21 29 58 28 42 58 21 30 20 45 54 59    001FF
                  55 54 41 54 53 3E 21 20 3C 39 0020E
00 29 58 28 42 58 21 31 20 45 54 59 42 20 53    00218          .ASCII  \S BYTE 1!XB(X)\<0><0>
                              00 00227
                                             010E0036  00228 P.AAW:  .LONG   17694774
                                             00000000' 0022C          .ADDRESS P.AAX
45 20 59 52 4F 4D 45 4D 3E 21 20 3C 39 33 21    00230 P.AAZ:  .ASCII  \!39< !>MEMORY EXTENSION BITS = !XB(X)!/!\
3D 20 53 54 49 42 20 4E 4F 49 53 4E 45 54 58    0023F
                  21 2F 21 29 58 28 42 58 21 20 0024E
42 4D 55 4E 20 45 4E 49 4C 3E 21 20 3C 39 33    00258          .ASCII  \39< !>LINE NUMBER = !ZB.\
                  2E 42 5A 21 20 3D 20 52 45    00267
                                             010E0040  00270 P.AAY:  .LONG   17694784
                                             00000000' 00274          .ADDRESS P.AAZ
54 45 4C 50 4D 4F 43 20 44 4E 41 4D 4D 4F 43    00278 P.ABB:  .ASCII  \COMMAND COMPLETED SUCCESSFULLY:!/:39< !>\
59 4C 4C 55 46 53 53 45 43 43 55 53 20 44 45    00287
                  3E 21 20 3C 39 33 21 2F 21 3A 00296
4C 42 20 43 41 21 20 44 45 56 49 45 43 45 52    002A0          .ASCII  \RECEIVED !AC BLOCK OF A!/!39< !>!AC MSG-
20 3C 39 33 21 2F 21 41 20 46 4F 20 4B 43 4F    002AF                  \<0>
                  00 47 53 4D 20 43 41 21 3E 21 002BE
                                             010E004F  002C8 P.ABA:  .LONG   17694799
                                             00000000' 002CC          .ADDRESS P.ABB
54 45 4C 50 4D 4F 43 20 44 4E 41 4D 4D 4F 43    002D0 P.ABD    .ASCII  \COMMAND COMPLETED SUCCESSFULLY:!/!39< !>\
59 4C 4C 55 46 53 53 45 43 43 55 53 20 44 45    002DF
                  3E 21 20 3C 39 33 21 2F 21 3A 002EE
55 42 20 43 41 21 20 44 45 56 49 45 43 45 52    002F8          .ASCII  \RECEIVED !AC BUFFER!/!39< !>!AC\<0>
41 21 3E 21 20 3C 39 33 21 2F 21 52 45 46 46    00307
                              00 43             00316
                                             010E0047  00318 P.ABC:  .LONG   17694791
                                             00000000' 0031C          .ADDRESS P.ABD
20 54 55 50 4E 49 20 52 45 54 53 49 47 45 52    00320 P.ABF:  .ASCII  \REGISTER INPUT TO KMS\
                              53 4D 4B 20 4F 54 0032F
                                             00335          .BLKB   3
                                             00000015  00338 P.ABE:  .LONG   21
                                             00000000' 0033C          .ADDRESS P.ABF
49 4E 55 20 4C 52 54 4E 43 20 4D 45 44 4F 4D    00340 P.ABH:  .ASCII  \MODEM CNTRL UNIT ADDRS = \<0><0><0>
   00 00 00 20 3D 20 53 52 44 44 41 20 54 52    0034F
                                             010E0019  0035C P.ABG:  .LONG   17694745
```

```
                                                00000000'  00360          .ADDRESS P.ABH
41 20 52 45 46 46 55 42 20 59 52 4F 4D 45 4D    00364 P.ARJ:   .ASCII  \MEMORY BUFFER ADDRS = \<0><0>
                  00 00 20 3D 20 53 52 44 44    00373
                                                010E0016   0037C P.ABI:   .LONG   17694742
                                                00000000'  00380          .ADDRESS P.ABJ
                                 54 53 52 49 46 05 00384 P.ABK:   .ASCII  <5>\FIRST\
                                    54 53 41 4C 04 0038A P.ABL:   .ASCII  <4>\LAST\
      45 54 41 49 44 45 4D 52 45 54 4E 49 0C    0038F P.ABM:   .ASCII  <12>\INTERMEDIATE\
                                    59 4C 4E 4F 04 0039C P.ABN:   .ASCII  <4>\ONLY\
54 55 50 54 55 4F 20 52 45 54 53 49 47 45 52    003A1 P.ABP:   .ASCII  \REGISTER OUTPUT FROM KMS\
                  53 4D 4B 20 4D 4F 52 46 20    003B0
                                                003B9          .BLKB   3
                                                00C00018   003BC P.ABO:   .LONG   24
                                                00000000'  003C0          .ADDRESS P.ABP
20 53 55 54 41 54 53 20 4E 57 4F 4E 4B 4E 55    003C4 P.ABR:   .ASCII  \UNKNOWN STATUS CODE = !XB(X)\
      29 58 28 42 58 21 20 3D 20 45 44 4F 43    003D3
                                                0000001C   003E0 P.ABQ:   .LONG   28
                                                00000000'  003E4          .ADDRESS P.ABR
5A 49 4C 41 49 54 49 4E 49 20 4D 45 44 4F 4D    003E8 P.ABS:   .ASCII  \MODEM INITIALIZATION\
                                    4E 4F 49 54 41 003F7
41 5A 49 4C 41 49 54 49 4E 49 20 45 4E 49 4C    003FC P.ABT:   .ASCII  \LINE INITIALIZATION\<0>
                                    00 4E 4F 49 54 0040B
49 4C 41 49 54 49 4E 49 45 44 20 45 4E 49 4C    00410 P.ABU:   .ASCII  \LINE DEINITIALIZATION\<0><0><0>
                  00 00 00 4E 4F 49 54 41 5A    0041F
50 49 52 43 53 45 44 20 53 53 45 52 44 44 41    00428 P.ABV:   .ASCII  \ADDRESS DESCRIPTOR CONNECT\<0><0>
   00 00 54 43 45 4E 4E 4F 43 20 52 4F 54 00437
50 49 52 43 53 45 44 20 53 53 45 52 44 44 41    00444 P.ABW:   .ASCII  \ADDRESS DESCRIPTOR DISCONNECT\<0><0>
00 54 43 45 4E 4E 4F 43 53 49 44 20 52 4F 54    00453
                                       00 00462
                                       00 00463          .ASCII  <0>
42 20 54 49 4D 53 4E 41 52 54 20 45 4E 49 4C    00464 P.ABX:   .ASCII  \LINE TRANSMIT BUFFER\
                                    52 45 46 46 55 00473
55 42 20 45 56 49 45 43 45 52 20 45 4E 49 4C    00478 P.ABY:   .ASCII  \LINE RECEIVE BUFFER\<0>
                                    00 52 45 46 46 00487
41 20 54 49 4D 53 4E 41 52 54 20 45 4E 49 4C    0048C P.ABZ:   .ASCII  \LINE TRANSMIT ABORT\<0>
                                    00 54 52 4F 42 0049B
42 41 20 45 56 49 45 43 45 52 20 45 4E 49 4C    004A0 P.ACA:   .ASCII  \LINE RECEIVE ABORT\<0><0>
                                    00 00 54 52 4F 004AF
53 43 49 54 53 49 54 41 54 53 20 45 4E 49 4C    004B4 P.ACB:   .ASCII  \LINE STATISTICS REQUEST\<0>
                  00 54 53 45 55 51 45 52 20    004C3
43 20 44 41 45 52 20 54 49 4D 53 4E 41 52 54    004CC P.ACC:   .ASCII  \TRANSMIT READ COMMAND\<0><0><0>
                  00 00 00 44 4E 41 4D 4D 4F    004DB
53 55 54 41 54 53 20 54 49 4D 53 4E 41 52 54    004E4 P.ACD:   .ASCII  \TRANSMIT STATUS\<0>
                                       00 004F3
4C 4C 4F 52 54 4E 4F 43 20 53 55 42 49 4E 55    004F4 P.ACE:   .ASCII  \UNIBUS CONTROLLER HAS STOPPED !/!39< !>A\
20 44 45 50 50 4F 54 53 20 53 41 48 20 52 45    00503
                  41 3E 21 20 3C 39 33 21 2F 21 00512
4E 41 4D 4D 4F 43 20 47 4E 49 54 50 45 43 43    0051C          .ASCII  \CCEPTING COMMANDS\<0><0><0>
                                    00 00 00 53 44 00528
20 44 45 55 53 53 49 20 44 4E 41 4D 4D 4F 43    00530 P.ACF:   .ASCII  \COMMAND ISSUED FOR THE LINE !/!39<  >NEV\
2F 21 20 45 4E 49 4C 20 45 48 54 20 52 4F 46    0053F
                  56 45 4E 3E 21 20 3C 39 33 21 0054E
                  44 45 54 45 4C 50 4D 4F 43 20 52 45 00558          .ASCII  \ER COMPLETED\
4F 54 20 44 45 4C 49 41 46 20 45 54 49 52 57    00564 P.ACG:   .ASCII  \WRITE FAILED TO COMPLETE IN THE !/!39< !\
48 54 20 4E 49 20 45 54 45 4C 50 4D 4F 43 20    00573
                  21 20 3C 39 33 21 2F 21 20 45 00582
                  00 00 00 45 4D 49 54 20 58 41 4D 3E 0058C          .ASCII  \>MAX TIME\<0><0><0>
```

```
20 45 44 4F 43 4F 52 43 49 4D 20 44 41 4F 4C 00598 P.ACH:  .ASCII  \LOAD MICROCODE FAILED\<0><0><0>
                  00 00 00 44 45 4C 49 41 46 005A7
45 43 49 56 45 44 20 4C 45 56 49 45 43 45 52 005B0 P.ACI:  .ASCII  \RECEIVED DEVICE STATUS MESSAGE\<0><0>
45 47 41 53 53 45 4D 20 53 55 54 41 54 53 20 005BF
                                    00 00 005CE
45 4C 49 41 46 20 45 44 4F 43 4F 52 43 49 4D 005D0 P.ACJ:  .ASCII  \MICROCODE FAILED TO START\<0><0><0>
      00 00 00 54 52 41 54 53 20 4F 54 20 44 005DF
20 44 4E 41 20 44 45 54 43 45 50 58 45 4E 55 005EC P.ACK:  .ASCII  \UNEXPECTED AND INCONSISTENT EVENT!/!39< \
56 45 20 54 4E 45 54 53 49 53 4E 4F 43 4E 49 005FB
                  20 3C 39 33 21 2F 21 54 4E 45 0060A
44 20 54 52 4F 50 20 44 45 53 55 41 43 3E 21 00614          .ASCII  \!>CAUSED PORT DRIVER TO SHUT LINE !/!39<\
4C 20 54 55 48 53 2C 4F 54 20 52 45 56 49 52 00623
                  3C 39 33 21 2F 21 20 45 4E 49 00632
                        00 4E 57 4F 44 3E 21 20 0063C          .ASCII  \ !>DOWN\<0>
20 50 41 4D 20 48 47 55 4F 4E 45 20 54 4F 4E 00644 P.ACL:  .ASCII  \NOT ENOUGH MAP REGISTERS !/!39< !>AVAILA\
39 33 21 2F 21 20 53 52 45 54 53 49 47 45 52 00653
                  41 4C 49 41 56 41 3E 21 20 3C 00662
45 54 49 52 57 20 41 20 52 4F 46 20 45 4C 42 0066C          .ASCII  \BLE FOR A WRITE\<0>
                                          00 0067B
49 45 43 45 52 20 53 45 54 59 42 20 53 2F 53 0067C P.ACM:  .ASCII  \S/S BYTES RECEIVED FROM CLASS !/!39< !>I\
20 53 53 41 4C 43 20 4D 4F 52 46 20 44 45 56 0068B
                  49 3E 21 20 3C 39 33 21 2F 21 0069A
00 41 54 41 44 20 46 4F 20 44 41 45 54 53 4E 006A4          .ASCII  \NSTEAD OF DATA\<0><0>
                                          00 006B3
44 45 56 49 45 43 45 52 20 53 55 54 41 54 53 006B4 P.ACN:  .ASCII  \STATUS RECEIVED FOR A LUN THAT !/!39< !>\
54 41 48 54 20 4E 55 4C 20 41 20 52 4F 46 20 006C3
                  3E 21 20 3C 39 33 21 2F 21 20 006D2
52 49 46 20 45 48 54 20 4F 53 4C 41 20 53 49 006DC          .ASCII  \IS ALSO THE FIRST LUN QUEUED TO !/!39< !\
54 20 44 45 55 45 55 51 20 4E 55 4C 20 54 53 006EB
                  21 20 3C 39 33 21 2F 21 20 4F 006FA
54 53 4F 48 28 20 54 49 4D 53 4E 41 52 54 3E 00704          .ASCII  \>TRANSMIT (HOST ONLY)\<0><0><0>
                  00 00 00 29 59 4C 4E 4F 20 00713
49 54 20 53 41 48 20 54 53 4F 48 20 4D 42 49 0071C P.ACO:  .ASCII  \IBM HOST HAS TIMED OUT !/!39< !>(SLAVE M\
20 3C 39 33 21 2F 21 20 54 55 4F 20 44 45 4D 0072B
                  4D 20 45 56 41 4C 53 28 3E 21 0073A
            00 00 00 29 59 4C 4E 4F 20 45 44 4F 00744          .ASCII  \ODE ONLY)\<0><0><0>
20 4E 4F 49 54 41 54 53 20 45 54 4F 4D 45 52 00750 P.ACP:  .ASCII  \REMOTE STATION COULD NOT BE !/!39< !>ADD\
2F 21 20 45 42 20 54 4F 4E 20 44 4C 55 4F 43 0075F
                  44 44 41 3E 21 20 3C 39 33 21 0076E
                        00 00 44 45 53 53 45 52 00778          .ASCII  \RESSED\<0><0>
20 4E 4F 49 54 41 54 53 20 45 54 4F 4D 45 52 00780 P.ACQ:  .ASCII  \REMOTE STATION ABORTED A BUFFER !/!39< !\
45 46 46 55 42 20 41 20 44 45 54 52 4F 42 41 0078F
                  21 20 3C 39 33 21 2F 21 20 52 0079E
                        29 45 56 41 4C 53 28 3E 007A8          .ASCII  \>(SLAVE)\
54 45 4D 41 52 41 50 20 44 49 4C 41 56 4E 49 007B0 P.ACR:  .ASCII  \INVALID PARAMETERS\<0><0>
                              00 00 53 52 45 007BF
45 43 45 52 20 48 47 55 4F 4E 45 20 54 4F 4E 007C4 P.ACS:  .ASCII  \NOT ENOUGH RECEIVE BUFFERS !/!39< !>PRES\
21 2F 21 20 53 52 45 46 46 55 42 20 45 56 49 007D3
                  53 45 52 50 3E 21 20 3C 39 33 007E2
                              00 54 4E 45 007EC          .ASCII  \ENT\<0>
4C 49 41 46 20 52 45 4C 4F 52 54 4E 4F 43 007F0 P.ACT:  .ASCII  \CONTROLLER FAILED TO ACCEPT !/!39< !>A C\
2F 21 20 54 50 45 43 41 20 4F 54 20 44 45 007FF
                  43 20 41 3E 21 20 3C 39 33 21 0080E
                        00 00 44 4E 41 4D 4D 4F 00818          .ASCII  \OMMAND\<0><0>
45 56 4F 20 45 55 45 55 51 20 4D 45 44 4F 4D 00820 P.ACU:  .ASCII  \MODEM QUEUE OVERFLOW\
                              57 4F 4C 46 52 0082F
46 52 45 56 4F 20 45 55 45 55 51 20 52 53 43 00834 P.ACV:  .ASCII  \CSR QUEUE OVERFLOW\<0><0>
```

```
                                                  00 00 57 4F 4C 00843
5A 49 53 20 45 47 41 53 53 45 4D 20 58 41 4D 00848 P.ACW:  .ASCII  \MAX MESSAGE SIZE EXCEEDED\<0><0><0>
   00 00 00 44 45 44 45 45 43 58 45 20 45 00857
20 53 49 48 54 20 52 4F 46 20 4D 45 44 4F 4D 00864 P.ACX:  .ASCII  \MODEM FOR THIS PORT FAILED\<0><0>
   00 00 44 45 4C 49 41 46 20 54 52 4F 50 00873
55 20 4C 4F 52 54 4E 4F 43 20 4D 45 44 4F 4D 00880 P.ACY:  .ASCII  \MODEM CONTROL UNIT HAS FAILED !/!39< !>O\
20 44 45 4C 49 41 46 20 53 41 48 20 54 49 4E 0088F
                  4F 3E 21 20 3C 39 33 21 2F 21 0089E
      00 54 52 4F 50 20 53 49 48 54 20 4E 008A8         .ASCII  \N THIS PORT\<0>
55 20 4C 4F 52 54 4E 4F 43 20 4D 45 44 4F 4D 008B4 P.ACZ:  .ASCII  \MODEM CONTROL UNIT DOES NO\ !/!39< !>RES\
2F 21 20 54 4F 4E 20 53 45 4F 44 20 54 49 4E 008C3
                  53 45 52 3E 21 20 3C 39 33 21 008D2
                              44 4E 4F 50 008DC         .ASCII  \POND\
52 20 46 4F 20 52 45 42 4D 55 4E 20 58 41 4D 008E0 P.ADA:  .ASCII  \MAX NUMBER OF RETRIES WAS !/!39< !>EXCEE\
33 21 2F 21 20 53 41 57 20 53 45 49 52 54 45 008EF
                  45 45 43 58 45 3E 21 20 3C 39 008FE
                              00 44 45 44 00908         .ASCII  \DED\<0>
44 4E 41 20 52 45 56 49 52 44 20 54 52 4F 50 0090C P.ADB:  .ASCII  \PORT DRIVER AND FIRMWARE ARE !/!39< !>OU\
21 20 45 52 41 20 45 52 41 57 4D 52 49 46 20 0091B
                  55 4F 3E 21 20 3C 39 33 21 2F 0092A
      00 00 00 43 4E 59 53 20 46 4F 20 54 00934         .ASCII  \; OF SYNC\<0><0><0>
45 56 4F 20 52 45 46 46 55 42 20 44 41 45 52 00940 P.ADC:  .ASCII  \READ BUFFER OVERFLOWED\<0><0>
                  00 00 44 45 57 4F 4C 46 52 0094F
3A 54 43 45 4A 45 52 20 44 4E 41 4D 4D 4F 43 00958 P.ADD:  .ASCII  \COMMAND REJECT: BUFFERS PENDING\<0>
4E 49 44 4E 45 50 20 53 52 45 46 46 55 42 20 00967
                                    00 47 00976
54 41 52 45 50 4F 20 59 4E 41 4D 20 4F 4F 54 00978 P.ADE:  .ASCII  \TOO MANY OPERATIONS OUTSTANDING\<0>
4E 49 44 4E 41 54 53 54 55 4F 20 53 4E 4F 49 00987
                                    00 47 00996
45 4D 20 54 4E 45 54 53 49 58 45 2D 4E 4F 4E 00998 P.ADF:  .ASCII  \NON-EXISTENT MEMORY ERROR\<0><0><0>
   00 00 00 52 4F 52 52 45 20 59 52 4F 4D 009A7
52 20 53 55 54 41 54 53 20 45 43 49 56 45 44 009B4 P.ADG:  .ASCII  \DEVICE STATUS RETURNED IN SEL 4 !/!39< !\
20 4C 45 53 20 4E 49 20 44 45 4E 52 55 54 45 009C3
                  21 20 3C 39 33 21 2F 21 20 34 009D2
      00 00 29 53 45 54 59 42 20 32 28 3E 009DC         .ASCII  \>(2 BYTES)\<0><0>
00 54 55 4F 20 44 45 4D 49 54 20 54 53 4F 48 009E8 P.ADH:  .ASCII  \HOST TIMED OUT\<0><0>
                                    00 009F7
49 4C 41 56 4E 49 20 3A 4E 55 4C 20 44 41 42 009F8 P.ADI:  .ASCII  \BAD LUN: INVALID LUN SPECIFIED\<0><0>
44 45 49 46 49 43 45 50 53 20 4E 55 4C 20 44 00A07
                                    00 00 00A16
55 42 20 54 58 45 54 20 44 45 54 52 4F 42 41 00A18 P.ADJ:  .ASCII  \ABORTED TEXT BUFFER: BUFFER !/!39< !>TER\
2F 21 20 52 45 46 46 55 42 20 3A 52 45 46 46 00A27
                  52 45 54 3E 21 20 3C 39 33 21 00A36
45 22 20 48 54 49 57 20 44 45 54 41 4E 49 4D 00A40         .ASCII  \MINATED WITH ''ENQ''\<0><0>
                              00 00 22 51 4E 00A4F
54 45 4D 41 52 41 50 20 44 49 4C 41 56 4E 49 00A54 P.ADK:  .ASCII  \INVALID PARAMETERS\<0><0>
                              00 00 53 52 45 00A63
45 43 45 52 20 48 47 55 4F 4E 45 20 54 4F 4E 00A68 P.ADL:  .ASCII  \NOT ENOUGH RECEIVE BUFFERS: !/!39< !>MES\
2F 21 20 3A 53 52 45 46 46 55 42 20 45 56 49 00A77
                  53 45 4D 3E 21 20 3C 39 33 21 00A86
45 43 45 52 20 47 4E 49 45 42 20 45 47 41 53 00A90         .ASCII  \SAGE BEING RECEIVED FOR WHICH !/!39< !>T\
20 48 43 49 48 57 20 52 4F 46 20 44 45 56 49 00A9F
                  54 3E 21 20 3C 39 33 21 2F 21 00AAE
45 52 20 4F 4E 20 45 52 45 57 20 45 52 45 48 00AB8         .ASCII  \HERE WERE NO RECEIVE BUFFERS\
      53 52 45 46 46 55 42 20 45 56 49 43 00AC7
20 44 45 49 46 49 43 45 50 53 20 52 45 53 55 00AD4 P.ADM:  .ASCII  \USER SPECIFIED AN INVALID !/!39< !>COMMA\
33 21 2F 21 20 44 49 4C 41 56 4E 49 20 4E 41 00AE3
```

```
                                41 4D 4D 4F 43 3E 21 20 3C 39  00AF2
4E 49 20 54 4F 4E 20 45 4E 49 4C 3A 44 4E  00AFC            .ASCII  \ND: LINE NOT IN SLAVE MODE\<0><0>
      00 00 45 44 4F 4D 20 45 56 41 4C 53 20  00B0B
45 56 4F 20 45 55 45 55 51 20 4D 45 44 4F 4D  00B18 P.ADN:   ASCII  \MODEM QUEUE OVERFLOW: COMMAND !/!39< !>A\
20 44 4E 41 4D 4D 4F 43 20 3A 57 4F 4C 46 52  00B27
                41 3E 21 20 3C 39 33 21 2F 21  00B36
53 20 4F 54 20 45 55 44 20 44 45 54 52 4F 42  00B40            .ASCII  \BORTED DUE TO SYSTEM BACKLOG\
      47 4F 4C 4B 43 41 42 20 4D 45 54 53 59  00B4F
46 52 45 56 4F 20 45 55 45 55 51 20 52 53 43  00B5C P.ADO:   .ASCII  \CSR QUEUE OVERFLOW (32 ENTRIES): !/!39< \
53 45 49 52 54 4E 45 20 32 33 28 20 57 4F 4C  00B6B
                20 3C 39 33 21 2F 21 20 3A 29  00B7A
20 50 45 46 20 4C 41 4E 52 45 54 4E 49 3E 21  00B84            .ASCII  \!>INTERNAL FEP DATA STRUCTURE !/!39< !>O\
20 45 52 55 54 43 55 52 54 53 2C 41 54 41 44  00B93
                4F 3E 21 20 3C 39 33 21 2F 21  00BA2
                      00 57 4F 4C 46 52 45 56  00BAC            .ASCII  \VERFLOW\<0>
59 54 50 4D 45 28 20 52 4F 52 52 45 20 31 4E  00BB4 P.ADP:   .ASCII  \N1 ERROR (EMPTY BUFFER OR ODD !/!39< !>A\
20 44 44 4F 20 52 4F 20 52 45 46 46 55 42 20  00BC3
                41 3E 21 20 3C 39 33 21 2F 21  00BD2
58 45 20 54 4E 55 4F 43 20 53 53 45 52 44 44  00BDC            .ASCII  \DDRESS COUNT EXCEEDED)\<0><0>
                00 00 29 44 45 44 45 45 43  00BEB
52 45 50 20 3A 4E 57 4F 44 20 4D 45 44 4F 4D  00BF4 P.ADQ:   .ASCII  \MODEM DOWN: PERIODIC CHECK OF !/!39< !>M\
20 46 4F 20 4B 43 45 48 43 20 43 49 44 4F 49  00C03
                4D 3E 21 20 3C 39 33 21 2F 21  00C12
44 20 2C 53 54 43 20 44 4E 55 4F 46 20 55 43  00C1C            .ASCII  \CU FOUND CTS, DSR, OR CD CLEARED\
52 41 45 4C 43 20 44 43 20 52 4F 20 2C 52 53  00C2B
                                44 45  00C3A
55 20 4C 4F 52 54 4E 4F 43 20 4D 45 44 4F 4D  00C3C P.ADR:   .ASCII  \MODEM CONTROL UNIT DOWN: !/!39< !>PERIOD\
39 33 21 2F 21 20 3A 4E 57 4F 44 20 54 49 4E  00C4B
                44 4F 49 52 45 50 3E 21 20 3C  00C5A
55 43 4D 20 46 4F 20 4B 43 45 48 43 20 43 49  00C64            .ASCII  \IC CHECK OF MCU FOUND RTS !/!39< !>OR DT\
33 21 2F 21 20 53 54 52 20 44 4E 55 4F 46 20  00C73
                54 44 20 52 4F 3E 21 20 3C 39  00C82
          00 00 00 44 45 52 41 45 4C 43 20 52  00C8C            .ASCII  \R CLEARED\<0><0><0>
4F 4D 20 54 4E 45 54 53 49 58 45 2D 4E 4F 4E  00C98 P.ADS:   .ASCII  \NON-EXISTENT MODEM CONTROL UNIT\<0>
49 4E 55 20 4C 4F 52 54 4E 4F 43 20 4D 45 44  00CA7
                                00 54  00CB6
54 46 41 20 45 53 4E 4F 50 53 45 52 20 4F 4E  00CB8 P.ADT:   .ASCII  \NO RESPONSE AFTER N2 RETRIES\
      53 45 49 52 54 45 52 20 32 4E 20 52 45  00CC7
57 4F 4C 46 52 45 56 4F 20 52 45 46 46 55 42  00CD4 P.ADU:   .ASCII  \BUFFER OVERFLOW: BUFFER RECEIVED !/!39< \
56 49 45 43 45 52 20 52 45 46 46 55 42 20 3A  00CE3
                20 3C 39 33 21 2F 21 20 44 45  00CF2
48 54 20 52 45 47 52 41 4C 20 53 41 57 3E 21  00CFC            .ASCII  \!>WAS LARGER THAN THE RECEIVE !/!39< !>B\
20 45 56 49 45 43 45 52 20 45 48 54 20 4E 41  00D0B
                42 3E 21 20 3C 39 33 21 2F 21  00D1A
44 45 54 41 43 4F 4C 4C 41 20 52 45 46 46 55  00D24            .ASCII  \UFFER ALLOCATED\<0>
                                00  00D33
3A 52 45 46 46 55 42 20 47 4E 49 44 4E 45 50  00D34 P.ADV:   .ASCII  \PENDING BUFFER: FEP REJECTED !/!39< !>CO\
21 20 44 45 54 43 45 4A 45 52 20 50 45 46 20  00D43
                4F 43 3E 21 20 3C 39 33 21 2F  00D52
54 20 45 53 55 41 43 45 42 20 44 4E 41 4D 4D  00D5C            .ASCII  \MMAND BECAUSE TRANSMIT OR !/!39< !>RECEI\
33 21 2F 21 20 52 4F 20 54 49 4D 53 4E 41 52  00D6B
                49 45 43 45 52 3E 21 20 3C 39  00D7A
45 52 45 57 20 53 52 45 46 46 55 42 20 45 56  00D84            .ASCII  \VE BUFFERS WERE OUTSTANDING\<0>
      00 47 4E 49 44 4E 41 54 53 54 55 4F 20  00D93
52 45 46 46 55 42 20 59 4E 41 4D 20 4F 4F 54  00DA0 P.ADW:   .ASCII  \TOO MANY BUFFERS: THIS BUFFER !/!39< !>W\
20 52 45 46 46 55 42 20 53 49 48 54 20 3A 53  00DAF
                57 3E 21 20 3C 39 33 21 2F 21  00DBE
```

```
                                                00DC8          .ASCII  \OULD CAUSE FEP TO HAVE 9 BUFFERS\
                                                00DD7
                                                00DE6
                                                00DE8 P.ADX:   .ASCII  \NON-EXISTENT MEMORY WHILE!/!39< !>ACCESS\
                                                00DF7
                                                00E06
                                                00E10          .ASCII  \ING !AC BUFFER\<0><0>
                                                00E1F
                                                00E20 P.ADY:   .ASCII  \OFF\<0>
                                                00E24 P.ADZ:   .ASCII  \RESOURCES ALLOCATED, NO REQUESTS !/!39< \
                                                00E33
                                                00E42
                                                00E4C          .ASCII  \!>ISSUED TO KMS\<0>
                                                00E5B
                                                00E5C P.AEA:   .ASCII  \TRANSITIONING UP/ATTEMPTING KMS !/!39< !\
                                                00E6B
                                                00E7A
                                                00E84          .ASCII  \>''INIT'' OR DOWN/ATTEMPTING !/!39< !>KMS \
                                                00E93
                                                00EA2
                                                00EAC          .ASCII  \''DEINIT''\
                                                00EB4 P.AEB:   .ASCII  \TRANSITIONING UP/ATTEMPTING !/!39< !>''CO\
                                                00EC3
                                                00ED2
                                                00EDC          .ASCII  \NNECT'' OR !/!39< !>DOWN/ATTEMPTING !/!39\
                                                00EEB
                                                00EFA
                                                00F04          .ASCII  \< !>''DISCONNECT''\
                                                00F13
                                                00F14 P.AEC:   .ASCII  \ACTIVE: MODEM OR MODEM CONTROL !/!39< !>\
                                                00F23
                                                00F32
                                                00F3C          .ASCII  \FAILURE\<0>
                                                00F44 P.AED:   .ASCII  \SHUTTING DOWN: ''TRANSMIT ABORT''!/!39< !>\
                                                00F53
                                                00F62
                                                00F6C          .ASCII  \ ISSUED\<0>
                                                00F74 P.AEE:   .ASCII  \SHUTTING DOWN: ''RECEIVE ABORT'' !/!39< !>\
                                                00F83
                                                00F92
                                                00F9C          .ASCII  \ISSUED\<0><0>
                                                00FA4 P.AEF:   .ASCII  \ACTIVE\<0><0>
                                                00FAC P.AEG:   .ASCII  \INTERRUPT ENABLE IN\<0>
                                                00FBB
                                                00FC0 P.AEH:   .ASCII  \INTERRUPT ENABLE OUT\
                                                00FCF
                                                00FD4 P.AEI:   .ASCII  \REQUEST IN\<0><0>
                                                00FE0 P.AEJ:   .ASCII  \STEP UP\<0>
                                                00FE8 P.AEK:   .ASCII  \RAM IN\<0><0>
                                                00FF0 P.AEL:   .ASCII  \RAM OUT\<0>
                                                00FF8 P.AEM:   .ASCII  \CRAM WRITE\<0><0>
                                                01004 P.AEN:   .ASCII  \MASTER CLEAR\
                                                01010 P.AEO:   .ASCII  \RUN\<0>
                                                01014 P.AEP:   .ASCII  \READY IN\
                                                0101C P.AEQ:   .ASCII  \READY OUT\<0><0><0>
                                                01028 P.AER:   .ASCII  <13>\DRIVER STATUS\
                                                01036 P.AES:   .ASCII  <4>\DATA\
```

```
                                                  54  53  45  54  04  0103B P.AET:  .ASCII  <4>\TEST\
                                                  54  53  45  54  04  01040 P.AEU:  .ASCII  <4>\TEST\
                                                  54  53  45  54  04  01045 P.AEV:  .ASCII  <4>\TEST\
                                                  54  53  45  54  04  0104A P.AEW:  .ASCII  <4>\TEST\
                45  54  41  54  53  20  45  4E  49  4C  0A  0104F P.AEX:  .ASCII  <10>\LINE STATE\

                                                                      .PSECT  $OWN$,NOEXE,  PIC,2

                                                              00000 LSTLUN: .BLKB   4
                                                              00004 ARGLIST:.BLKB   40
30  2F  2E  2D  28  27  10  0E  0C  0A  09  08  06  04  02  0002C BTS_STS_CODES:
                                                                      .BYTE   2, 4, 6, 8, 9, 10, 12, 14, 16, 39, 40, -
        3F  3E  3D  3A  39  38  37  36  35  34  33  32  31  0003B                45, 46, 47, 48, 49, 50, 51, 52, 53, 54, -
                                                                                55, 56, 57, 58, 61, 62, 63
39  38  37  36  35  34  33  32  31  30  2F  2E  2D  28  09  00048 CMD_STS_CODES:
                                                                      .BYTE   9, 40, 45, 46, 47, 48, 49, 50, 51, 52, -
                                                  3F  3E  3D  3A  00057                53, 54, 55, 56, 57, 58, 61, 62, 63
                                                              0005B       .BLKB   1
                                                      0000E791  0005C CSR0_MASK:
                                                                      .LONG   59281
                                                      00000090  00060 CSR1_MASK:
                                                                      .LONG   144
                                                      00000000' 00064 DEV_ID_DESC:
                                                                      .ADDRESS P.AAA
                                                              00068 FAO_STRING:
                                                                      .BLKB   4
                                                      00000000' 0006C FAO_STRING1:
                                                                      .ADDRESS P.AAC
                                                      00000000' 00070 FAO_STRINGS:
                                                                      .ADDRESS P.AAE
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00074             .ADDRESS P.AAG, P.AAI, P.AAK, P.AAM, P.AAO, -
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 0008C                     P.AAQ, F.AAS, P.AAU, P.AAW, P.AAY, P.ABA, -
                                                                                      P.ABC
                                                      00000000' 000A4 INPUT_HEADER:
                                                                      .ADDRESS P.ABE
                                                      00000000' 000A8 MCU_STRING:
                                                                      .ADDRESS P.ABG
                                                      00000000' 000AC MEM_STRING:
                                                                      .ADDRESS P.ABI
                00000000' 00000000' 00000000' 00000000' 000B0 MSG:    .ADDRESS P.ABK, P.ABL, P.ABM, P.ABN
                                                      00000000  000C0 OUT_ARGLIST:
                                                                      .LONG   0
                                                      00000000' 000C4 OUTPUT_HEADER:
                                                                      .ADDRESS P.ABO
                                                              000C8 STATUS: .BLKB   4
                                                      00000000' 000CC UNKNOWN_DESC:
                                                                      .ADDRESS P.ABQ
                                                      00000014  000D0 KMSCMDS_DESC_TBL:
                                                                      .LONG   20
                                                      00000000' 000D4         .ADDRESS P.ABS
                                                      00000013  000D8         .LONG   19
                                                      00000000' 000DC         .ADDRESS P.ABT
                                                      00000015  000E0         .LONG   21
                                                      00000000' 000E4         .ADDRESS P.ABU
                                                      0000001A  000E8         .LONG   26
                                                      00000000' 000EC         .ADDRESS P.ABV
                                                      0000001D  000F0         .LONG   29
```

```
00000000' 000F4          .ADDRESS P.ABW
00000014  000F8          .LONG   20
00000000' 000FC          .ADDRESS P.ABX
00000013  00100          .LONG   19
00000000' 00104          .ADDRESS P.ABY
00000013  00108          .LONG   19
00000000' 0010C          .ADDRESS P.ABZ
00000012  00110          .LONG   18
00000000' 00114          .ADDRESS P.ACA
00000017  00118          .LONG   23
00000000' 0011C          .ADDRESS P.ACB
00000015  00120          .LONG   21
00000000' 00124          .ADDRESS P.ACC
0000000F  00128          .LONG   15
00000000' 0012C          .ADDRESS P.ACD
00000039  00130 BTSTATUS_DESC_TBL:
                         .LONG   57
00000000' 00134          .ADDRESS P.ACE
00000034  00138          .LONG   52
00000000' 0013C          .ADDRESS P.ACF
00000031  00140          .LONG   49
00000000' 00144          .ADDRESS P.ACG
00000015  00148          .LONG   21
00000000' 0014C          .ADDRESS P.ACH
0000001E  00150          .LONG   30
00000000' 00154          .ADDRESS P.ACI
00000019  00158          .LONG   25
00000000' 0015C          .ADDRESS P.ACJ
00000057  00160          .LONG   87
00000000' 00164          .ADDRESS P.ACK
00000037  00168          .LONG   55
00000000' 0016C          .ADDRESS P.ACL
00000036  00170          .LONG   54
00000000' 00174          .ADDRESS P.ACM
00000065  00178          .LONG   101
00000000' 0017C          .ADDRESS P.ACN
00000031  00180          .LONG   49
00000000' 00184          .ADDRESS P.ACO
0000002E  00188          .LONG   46
00000000' 0018C          .ADDRESS P.ACP
00000030  00190          .LONG   48
00000000' 00194          .ADDRESS P.ACQ
00000012  00198          .LONG   18
00000000' 0019C          .ADDRESS P.ACR
0000002B  001A0          .LONG   43
00000000' 001A4          .ADDRESS P.ACS
0000002E  001A8          .LONG   46
00000000' 001AC          .ADDRESS P.ACT
00000014  001B0          .LONG   20
00000000' 001B4          .ADDRESS P.ACU
00000012  001B8          .LONG   18
00000000' 001BC          .ADDRESS P.ACV
00000019  001C0          .LONG   25
00000000' 001C4          .ADDRESS P.ACW
0000001A  001C8          .LONG   26
00000000' 001CC          .ADDRESS P.ACX
00000033  001D0          .LONG   51
```

```
                        00000000' 001D4          .ADDRESS P.ACY
                        0000002C  001D8          .LONG   44
                        00000000' 001DC          .ADDRESS P.ACZ
                        0000002B  001E0          .LONG   43
                        00000000' 001E4          .ADDRESS P.ADA
                        00000031  001E8          .LONG   49
                        00000000' 001EC          .ADDRESS P.ADB
                        00000016  001F0          .LONG   22
                        00000000' 001F4          .ADDRESS P.ADC
                        0000001F  001F8          .LONG   31
                        00000000' 001FC          .ADDRESS P.ADD
                        0000001F  00200          .LONG   31
                        00000000' 00204          .ADDRESS P.ADE
                        00000019  00208          .LONG   25
                        00000000' 0020C          .ADDRESS P.ADF
                        00000032  00210 CMDSTS_DESC_TBL:
                                                 .LONG   50
                        00000000' 00214          .ADDRESS P.ADG
                        0000000E  00218          .LONG   14
                        00000000' 0021C          .ADDRESS P.ADH
                        0000001E  00220          .LONG   30
                        00000000' 00224          .ADDRESS P.ADI
                        0000003A  00228          .LONG   58
                        00000000' 0022C          .ADDRESS P.ADJ
                        00000012  00230          .LONG   18
                        00000000' 00234          .ADDRESS P.ADK
                        0000006C  00238          .LONG   108
                        00000000' 0023C          .ADDRESS P.ADL
                        00000042  00240          .LONG   66
                        00000000' 00244          .ADDRESS P.ADM
                        00000044  00248          .LONG   68
                        00000000' 0024C          .ADDRESS P.ADN
                        00000057  00250          .LONG   87
                        00000000' 00254          .ADDRESS P.ADO
                        0000003E  00258          .LONG   62
                        00000000' 0025C          .ADDRESS P.ADP
                        00000048  00260          .LONG   72
                        00000000' 00264          .ADDRESS P.ADQ
                        00000059  00268          .LONG   89
                        00000000' 0026C          .ADDRESS P.ADR
                        0000001F  00270          .LONG   31
                        00000000' 00274          .ADDRESS P.ADS
                        0000001C  00278          .LONG   28
                        00000000' 0027C          .ADDRESS P.ADT
                        0000005F  00280          .LONG   95
                        00000000' 00284          .ADDRESS P.ADU
                        0000006B  00288          .LONG   107
                        00000000' 0028C          .ADDRESS P.ADV
                        00000048  00290          .LONG   72
                        00000000' 00294          .ADDRESS P.ADW
                        00000036  00298          .LONG   54
                        00000000' 0029C          .ADDRESS P.ADX
                        00000003  002A0 LINE_STATES_DESC_TBL:
                                                 .LONG   3
                        00000000' 002A4          .ADDRESS P.ADY
                        00000037  002A8          .LONG   55
                        00000000' 002AC          .ADDRESS P.ADZ
```

```
                00000058  002B0          .LONG   88
                00000000' 002B4          .ADDRESS P.AEA
                00000060  002B8          .LONG   96
                00000000' 002BC          .ADDRESS P.AEB
                0000002F  002C0          .LONG   47
                00000000' 002C4          .ADDRESS P.AEC
                0000002F  002C8          .LONG   47
                00000000' 002CC          .ADDRESS P.AED
                0000002E  002D0          .LONG   46
                00000000' 002D4          .ADDRESS P.AEE
                00000006  002D8          .LONG   6
                00000000' 002DC          .ADDRESS P.AEF
                00000013  002E0 CSR0_DESC_TBL:
                                         .LONG   19
                00000000' 002E4          .ADDRESS P.AEG
                00000014  002E8          .LONG   20
                00000000' 002EC          .ADDRESS P.AEH
                0000000A  002F0          .LONG   10
                00000000' 002F4          .ADDRESS P.AEI
                0C000007  002F8          .LONG   7
                00000000' 002FC          .ADDRESS P.AEJ
                00000006  00300          .LONG   6
                00000000' 00304          .ADDRESS P.AEK
                00000007  00308          .LONG   7
                00000000' 0030C          .ADDRESS P.AEL
                0000000A  00310          .LONG   10
                00000000' 00314          .ADDRESS P.AEM
                0000000C  00318          .LONG   12
                00000000' 0031C          .ADDRESS P.AEN
                00000003  00320          .LONG   3
                00000000' 00324          .ADDRESS P.AEO
                00000008  00328 CSR1_DESC_TBL:
                                         .LONG   8
                00000000' 0032C          .ADDRESS P.AEP
                00000009  00330          .LONG   9
                00000000' 00334          .ADDRESS P.AEQ

                                         .EXTRN  EMB, SYECOM, FRCTOF
                                         .EXTRN  DHEAD1, ORB$L_OWNER_RTN
                                         .EXTRN  OUTPUT_LINES, TRANSLATE_BITS
                                         .EXTRN  UBA_MAPPING, UCB$L_CHAR
                                         .EXTRN  UCB$W_STS_RTN, UCB$L_OPCNT_RTN
                                         .EXTRN  UCB$W_ERRCNT_RTN
                                         .EXTRN  SYS$FAOL

                                         .PSECT  $CODE,NOWRT,  PIC,2

                          07FC 00000     .ENTRY  KMS3271, Save R2,R3,R4,R5,R6,R7,R8,R9,R10    ; 0655
5A 00000000G   00   9E 00002             MOVAB   UBA_MAPPING, R10
59 00000000V   00   9E 00009             MOVAB   OUTPUT_REGISTERS, R9
58 00000000G   00   9E 00010             MOVAB   OUTPUT_LINES, R8
57 00000000G   00   9E 00017             MOVAB   LIB$STOP, R7
56 00000000G   00   9E 0001E             MOVAB   SYS$FAOL, R6
55 00000000'   00   9E 00025             MOVAB   P.AER, R5
54 00000000G   00   9E 0002C             MOVAB   BTSTATUS, R4
53 00000000'   00   9E 00033             MOVAB   LSTLUN, R3
5E      FDF4   CE   9E 0003A             MOVAB   -524(SP), SP
```

```
                        04    AE 010E0000    8F   D0 0003F          MOVL    #17694720, DESC                                  ; 0691
                                     08    AE   D4 00047          CLRL    DESC+4                                           ;
                        04    AE       0200  8F   B0 0004A          MOVW    #512, DESC                                       ; 0699
                        08    AE         0C  AE   9E 00050          MOVAB   BUFFER, DESC+4                                   ; 0700
                              63 00000000G   00   D0 00055          MOVL    SYECOM+39, LSTLUN                                ; 0701
                                             53   DD 0005C          PUSHL   R3                                               ; 0706
                  00000000G   00             01   FB 0005E          CALLS   #1, FRCTOF                                       ;
                                       64    A3   DD 00065          PUSHL   DEV_ID_DESC                                      ; 0707
                                             53   DD 00068          PUSHL   R3                                               ;
                  00000000G   00             02   FB 0006A          CALLS   #2, DHEAD1                                       ;
                        04    A3             65   9E 00071          MOVAB   P.AER, ARGLIST                                   ; 0716
                              50       04    A3   D0 00075          MOVL    ARGLIST, R0                                      ; 0717
                        08    A3             60   9A 00079          MOVZBL  (R0), ARGLIST+4                                  ;
              08  A3              14       08    A3   C3 0007D          SUBL3   ARGLIST+4, #20, ARGLIST+4                        ;
                              52             64   9A 00083          MOVZBL  BTSTATUS, R2                                     ; 0718
                        0C    A3             52   D0 00086          MOVL    R2, ARGLIST+8                                    ;
                        10    A3       04    AE   9E 0008A          MOVAB   DESC, ARGLIST+12                                 ; 0719
                                             50   7C 0008F          CLRQ    INDEX                                            ; 0727
                              52       2C A340   91 00091 1$:       CMPB    BTS_STS_CODES[INDEX], R2                         ; 0729
                                       0C        12 00096          BNEQ    2$                                               ;
                        14    A3       0130 C340 7E 00098          MOVAQ   BTSTATUS_DESC_TBL[INDEX], ARGLIST+16             ; 0736
                              51             01   D0 0009F          MOVL    #1, FLAG                                         ; 0737
                                             04   11 000A2          BRB     3$                                               ; 0735
                  E9          50             1B   F3 000A4 2$:       AOBLEQ  #27, INDEX, 1$                                   ; 0727
                              03             51   E9 000A8 3$:       BLBC    FLAG, 4$                                         ; 0745
                                       0099  31   00AB          BRW     17$                                              ;
                        1C    A3       0E    A5   9E 000AE 4$:       MOVAB   P.AES, ARGLIST+24                                ; 0752
                              51             01   CE 000B3          MNEGL   #1, INDEX2                                       ; 0753
                        16    01             52   8F 000B6          CASEB   R2, #1, #22                                      ; 0755
              0068      0033      0068       002E      000BA 5$:       .WORD   6$-5$,-                                          ;
              0068      003C      0068       0037      000C2           .WORD   14$-5$,-                                         ;
              0068      0068      0068       0068      000CA           .WORD   7$-5$,-                                          ;
              0068      0068      0068       0068      000D2           .WORD   14$-5$,-                                         ;
              0068      004B      0068       0041      000DA           .WORD   8$-5$,-                                          ;
                        005E      0068       0054      000E2           .WORD   14$-5$,-                                         ;
                                                                       .WORD   9$-5$,-                                          ;
                                                                       .WORD   14$-5$,-                                         ;
                                                                       .WORD   14$-5$,-                                         ;
                                                                       .WORD   14$-5$,-                                         ;
                                                                       .WORD   14$-5$,-                                         ;
                                                                       .WORD   14$-5$,-                                         ;
                                                                       .WORD   14$-5$,-                                         ;
                                                                       .WORD   14$-5$,-                                         ;
                                                                       .WORD   14$-5$,-                                         ;
                                                                       .WORD   14$-5$,-                                         ;
                                                                       .WORD   10$-5$,-                                         ;
                                                                       .WORD   14$-5$,-                                         ;
                                                                       .WORD   11$-5$,-                                         ;
                                                                       .WORD   14$-5$,-                                         ;
                                                                       .WORD   12$-5$,-                                         ;
                                                                       .WORD   14$-5$,-                                         ;
                                                                       .WORD   13$-5$                                           ;
                              51             02   D0 000E8 6$:       MOVL    #2, INDEX2                                       ; 0757
                                             38   11 000EB          BRB     15$                                              ;
                                             51   D4 000ED 7$:       CLRL    INDEX2                                           ; 0758
                                             34   11 000EF          BRB     15$                                              ;
                              51             01   D0 000F1 8$:       MOVL    #1, INDEX2                                       ; 0759
```

```
                                  2F  11 000F4        BRB     15$                                              ;
                   51             03  D0 000F6  9$:   MOVL    #3, INDEX2                                       ; 0760
                                  2A  11 000F9        BRB     15$                                              ;
                   51             02  D0 000FB 10$:   MOVL    #2, INDEX2                                       ; 0763
             1C    A3        13   A5  9E 000FE        MOVAB   P.AET, ARGLIST+24                                ; 0764
                                  20  11 00103        BRB     15$                                              ; 0755
                                  51  D4 00105 11$:   CLRL    INDEX2                                           ; 0769
             1C    A3        18   A5  9E 00107        MOVAB   P.AEU, ARGLIST+24                                ; 0770
                                  17  11 0010C        BRB     15$                                              ; 0755
                   51             01  D0 0010E 12$:   MOVL    #1, INDEX2                                       ; 0775
             1C    A3        1D   A5  9E 00111        MOVAB   P.AEV, ARGLIST+24                                ; 0776
                                  0D  11 00116        BRB     15$                                              ; 0755
                   51             03  D0 00118 13$:   MOVL    #3, INDEX2                                       ; 0781
             1C    A3        22   A5  9E 0011B        MOVAB   P.AEW, ARGLIST+24                                ; 0782
                                  03  11 00120        BRB     15$                                              ; 0755
                   51             01  CE 00122 14$:   MNEGL   #1, INDEX2                                       ; 0786
       FFFFFFFF    8F             51  D1 00125 15$:   CMPL    INDEX2, #-1                                      ; 0792
                                  0F  13 0012C        BEQL    16$                                              ;
             14    A3      009C   C3  D0 0012E        MOVL    FAO_STRINGS+44, ARGLIST+16                       ; 0799
             18    A3      0080 C341  D0 00134        MOVL    MSG[INDEX2], ARGLIST+20                          ; 0800
                                  0A  11 0013B        BRB     17$                                              ; 0792
             14    A3      00CC   C3  D0 0013D 16$:   MOVL    UNKNOWN_DESC, ARGLIST+16                         ; 0808
             18    A3             52  D0 00143        MOVL    R2, ARGLIST+20                                   ; 0809
                             18   A3  9F 00147 17$:   PUSHAB  ARGLIST+20                                       ; 0816
                             08   AE  9F 0014A        PUSHAB  DESC                                             ;
                             0C   AE  9F 0014D        PUSHAB  DESC                                             ;
                             14   A3  DD 00150        PUSHL   ARGLIST+16                                       ;
                   66             04  FB 00153        CALLS   #4, SYS$FAOL                                     ;
           00C8    C3             50  D0 00156        MOVL    R0, STATUS                                       ;
                   07             50  E8 0015B        BLBS    R0, 18$                                          ;
                           00C8   C3  DD 0015E        PUSHL   STATUS                                           ;
                   67             01  FB 00162        CALLS   #1, LIB$STOP                                     ;
                             10   A3  9F 00165 18$:   PUSHAB  ARGLIST+12                                       ; 0819
                             7C   A3  DD 00168        PUSHL   FAO_STRINGS+12                                   ;
                             04   A3  9F 0016B        PUSHAB  ARGLIST                                          ; 0818
                             6C   A3  DD 0016E        PUSHL   FAO_STRING1                                      ;
                   68             04  FB 00171        CALLS   #4, OUTPUT_LINES                                 ;
             04    A3        27   A5  9E 00174        MOVAB   P.AEX, ARGLIST                                   ; 0828
                   50        04   A3  D0 00179        MOVL    ARGLIST, R0                                      ; 0829
             08    A3             60  9A 0017D        MOVZBL  (R0), ARGLIST+4                                  ;
 08    A3          14        08   A3  C3 00181        SUBL3   ARGLIST+4, #20, ARGLIST+4                        ;
                   50        04   A4  9A 00187        MOVZBL  LINE_STATE, R0                                   ; 0830
             0C    A3             50  D0 0018B        MOVL    R0, ARGLIST+8                                    ;
             10    A3      02A0 C340  7E 0018F        MOVAQ   LINE_STATES_DESC_TBL[R0], ARGLIST+12             ; 0831
                                  50  D5 00196        TSTL    R0                                               ; 0836
                                  05  12 00198        BNEQ    19$                                              ;
                   07             50  91 0019A        CMPB    R0, #7                                           ; 0837
                                  2A  13 0019D        BEQL    21$                                              ;
             04    AE      0200   8F  B0 0019F 19$:   MOVW    #512, DESC                                       ; 0843
                                  6E  D4 001A5        CLRL    (SP)                                             ; 0844
                                  5E  DD 001A7        PUSHL   SP                                               ;
                             08   AE  9F 001A9        PUSHAB  DESC                                             ;
                             0C   AE  9F 001AC        PUSHAB  DESC                                             ;
                             10   A3  DD 001AF        PUSHL   ARGLIST+12                                       ;
                   66             04  FB 001B2        CALLS   #4, SYS$FAOL                                     ;
           00C8    C3             50  D0 001B5        MOVL    R0, STATUS                                       ;
                   07             50  E8 001BA        BLBS    R0, 20$                                          ;
```

```
                                00C8   C3  DD 001BD          PUSHL   STATUS                                          ;
                                   67  01  FB 001C1          CALLS   #1, LIB$STOP                                    ;
                                10 A3    04  AE  9E 001C4 20$:  MOVAB   DESC, ARGLIST+12                            ; 0845
                                   10    A3  9F 001C9 21$:  PUSHAB  ARGLIST+12                                      ; 0852
                                   7C    A3  DD 001CC       PUSHL   FAO_STRINGS+12                                  ;
                                   04    A3  9F 001CF       PUSHAB  ARGLIST                                         ; 0851
                                   6C    A3  DD 001D2       PUSHL   FAO_STRING1                                     ;
                                      68     04  FB 001D5       CALLS   #4, OUTPUT_LINES                                ;
                                             7E  D4 001D8       CLRL    -(SP)                                           ; 0857
                                      69     01  FB 001DA       CALLS   #1, OUTPUT_REGISTERS                            ;
                                             01  DD 001DD       PUSHL   #1                                              ; 0862
                                      69     01  FB 001DF       CALLS   #1, OUTPUT_REGISTERS                            ;
                                      50     64  9A 001E2       MOVZBL  BTSTATUS, R0                                    ; 0868
                                      3F     50  91 001E5       CMPB    R0, #63                                         ;
                                             05  13 001E8       BEQL    22$                                             ;
                                      0E     50  91 001EA       CMPB    R0, #14                                         ; 0869
                                             1A  12 001ED       BNEQ    23$                                             ;
                                   2C    A4  9F 001EF 22$:  PUSHAB  MAP_CONTENTS                                    ; 0875
                                   28    A4  9F 001F2       PUSHAB  MAP_NUMBER                                      ;
                                         53  DD 001F5       PUSHL   R3                                              ;
                                      6A     03  FB 001F7       CALLS   #3, UBA_MAPPING                                 ;
                                   30    A4  9F 001FA       PUSHAB  FINAL_MAP_CONTENTS                              ; 0876
                                04 AE    01  CE 001FD       MNEGL   #1, 4(SP)                                       ;
                                   04    AE  9F 00201       PUSHAB  4(SP)                                           ;
                                         53  DD 00204       PUSHL   R3                                              ;
                                      6A     03  FB 00206       CALLS   #3, UBA_MAPPING                                 ;
                                   E0    A4  9F 00209 23$:  PUSHAB  EMB+50                                          ; 0883
                                         53  DD 0020C       PUSHL   R3                                              ;
                         00000000G  00          02  FB 0020E       CALLS   #2, ORB$L_OWNER_RTN                             ;
                                   E4    A4  9F 00215       PUSHAB  EMB+54                                          ; 0884
                                         53  DD 00218       PUSHL   R3                                              ;
                         00000000G  00          02  FB 0021A       CALLS   #2, UCB$L_CHAR                                  ;
                                   C8    A4  9F 00221       PUSHAB  EMB+26                                          ; 0885
                                         53  DD 00224       PUSHL   R3                                              ;
                         00000000G  00          02  FB 00226       CALLS   #2, UCB$W_STS_RTN                               ;
                                   DC    A4  9F 0022D       PUSHAB  EMB+46                                          ; 0886
                                         53  DD 00230       PUSHL   R3                                              ;
                         00000000G  00          02  FB 00232       CALLS   #2, UCB$L_OPCNT_RTN                             ;
                                   DA    A4  9F 00239       PUSHAB  EMB+44                                          ; 0887
                                         53  DD 0023C       PUSHL   R3                                              ;
                         00000000G  00          02  FB 0023E       CALLS   #2, UCB$W_ERRCNT_RTN                            ;
                                               04  00245       RET                                                     ; 0893

; Routine Size:  582 bytes,    Routine Base:  $CODE + 0000

;  480          0894  1
```

```
;  482          0895  1 Routine OUTPUT_REGISTERS (flag) : NOVALUE =
;  483          0896  2 Begin
;  484          0897  2
;  485          0898  2 !++
;  486          0899  2 !
;  487          0900  2 ! Functional Description:
;  488          0901  2 !
;  489          0902  2 !    This routine
;  490          0903  2 !
;  491          0904  2 ! Calling Sequence:
;  492          0905  2 !
;  493          0906  2 !    OUTPUT_REGISTERS (flag) ;
;  494          0907  2 !
;  495          0908  2 ! Input Parameters:
;  496          0909  2 !
;  497          0910  2 !    Flag = indicates whether the registers are input to or output from
;  498          0911  2 !           the KMS (0 = input to KMS,  1 = output from KMS).
;  499          0912  2 !
;  500          0913  2 ! Output Parameters:
;  501          0914  2 !
;  502          0915  2 !    None
;  503          0916  2 !
;  504          0917  2 !--
;  505          0918  2
;  506          0919  2 Bind regsav = (emb[emb$l_dv_regsav]+4) : vector [,lorg] ;
;  507          0920  2
;  508          0921  2 LOCAL
;  509          0922  2     Index,
;  510          0923  2     Index2,
;  511          0924  2     Reg_buffer: Vector [512,byte],
;  512          0925  2     Reg_desc:   $BBLOCK [8]
;  513          0926  2                 Preset ( [dsc$b_dtype] = dsc$k_dtype_t,
;  514          0927  2                          [dsc$b_class] = dsc$k_class_s,
;  515          0928  2                          [dsc$w_length] = 0,
;  516          0929  2                          [dsc$a_pointer] = reg_buffer ),
;  517          0930  2     Str_number ;
;  518          0931  2
;  519          0932  2 OWN
;  520          0933  2     Csr0:       REF $BBLOCK,
;  521          0934  2     Csr1:       REF $BBLOCK,
;  522          0935  2     Csr2:       REF $BBLOCK,
;  523          0936  2     Csr3:       REF $BBLOCK ;
;  524          0937  2
;  525          0938  2 !
;  526          0939  2 ! Determine whether to translate input or output CSR's.
;  527          0940  2 !
;  528          0941  2 If NOT .flag
;  529          0942  2 Then
;  530          0943  2     !
;  531          0944  2     ! Input to KMS, get the input registers and the header.
;  532          0945  2     !
;  533          0946  3     Begin
;  534          0947  3     Csr0 = regsav[2] ;
;  535          0948  3     Csr1 = regsav[3] ;
;  536          0949  3     Csr2 = regsav[4] ;
;  537          0950  3     Csr3 = regsav[5] ;
;  538          0951  3
```

```
;  539          0952  3     Arglist[0] = .input_header ;
;  540          0953  3     End
;  541          0954  2 Else
;  542          0955  2
;  543          0956  2     ! Output from KMS, get the output registers and the header.
;  544          0957  2     !
;  545          0958  3     Begin
;  546          0959  3     Csr0 = regsav[6] ;
;  547          0960  3     Csr1 = regsav[7] ;
;  548          0961  3     Csr2 = regsav[8] ;
;  549          0962  3     Csr3 = regsav[9] ;
;  550          0963  3
;  551          0964  3     Arglist[0] = .output_header ;
;  552          0965  2     End ;
;  553          0966  2
;  554          0967  2 !
;  555          0968  2 ! Output the header.
;  556          0969  2 !
;  557          0970  2 OUTPUT_LINES (%ASCID '!/!1< !>!AS',arglist[0]) ;
;  558          0971  2
;  559          0972  2 !
;  560          0973  2 ! Translate and output the registers.
;  561          0974  2 !
;  562          0975  2 ! CSR0
;  563          0976  2 !
;  564          0977  2 ! Arglist contains: the register mnemonic, the starting position of the
;  565          0978  2 ! register contents field, and the register contents
;  566          0979  2 !
;  567          0980  2 Arglist[0] = CSTRING ('CSR0') ;
;  568          0981  2 Arglist[1] = (16 - .(.arglist[0])<0,8>)+4 ;
;  569          0982  2 Arglist[2] = ..csr0 ;
;  570          0983  2
;  571          0984  2 !
;  572          0985  2 ! Translate bits to text for the csr0 bits defined by csr0_mask.
;  573          0986  2 !
;  574          0987  3 If ( status = TRANSLATE_BITS (.csr0,csr0_mask,csr0_desc_tbl,
;  575          0988  3                              out_arglist,fao_string,%REF(0)) )
;  576          0989  2 Then
;  577          0990  2     !
;  578          0991  2     ! Bits translated, output the csr0 information.
;  579          0992  2     !
;  580          0993  3     Begin
;  581          0994  3     OUTPUT_LINES ( .fao_string1,arglist[0],
;  582          0995  3                    .fao_string,out_arglist[0] ) ;
;  583          0996  3     End
;  584          0997  2 Else
;  585          0998  2     !
;  586          0999  2     ! No bits translated, output the csr0 information.
;  587          1000  2     !
;  588          1001  2     OUTPUT_LINES ( .fao_string1,arglist[0] ) ;
;  589          1002  2
;  590          1003  2
;  591          1004  2
;  592          1005  2 !
;  593          1006  2 ! CSR1
;  594          1007  2 !
;  595          1008  2 ! Arglist contains: the register mnemonic, the starting position of the
```

```
;  596          1009  2 ! register contents field, the register contents, and the address of the
;  597          1010  2 ! descriptor that contains the kms command text.
;  598          1011  2 !
;  599          1012  2 Arglist[0] = CSTRING ('CSR1') ;
;  600          1013  2 Arglist[1] = (16 - .(.arglist[0])<0,8>)+4 ;
;  601          1014  2 Arglist[2] = ..csr1 ;
;  602          1015  2 Arglist[3] = kmscmds_desc_tbl[.csr1[cmd],0,0,0,0] ;
;  603          1016  2
;  604          1017  2 !
;  605          1018  2 ! Translate bits to text for the csr1 bits defined by csr1_mask.
;  606          1019  2 !
;  607          1020  3 If ( status = TRANSLATE_BITS (.csr1,csr1_mask,csr1_desc_tbl,
;  608          1021  3                              out_arglist,fao_string,%REF(0)) )
;  609          1022  2 Then
;  610          1023  2     !
;  611          1024  2     ! Bits translated, output the first part of the csr1 information.
;  612          1025  2     !
;  613          1026  3     Begin
;  614          1027  3     OUTPUT_LINES (  .fao_string1,arglist[0],
;  615          1028  3                     .fao_strings[3],arglist[3],
;  616          1029  3                     .fao_string,out_arglist[0] ) ;
;  617          1030  3     End
;  618          1031  2 Else
;  619          1032  2     !
;  620          1033  2     ! No bits translated, output the first part of the csr1 information.
;  621          1034  2     !
;  622          1035  2     OUTPUT_LINES (  .fao_string1,arglist[0],
;  623          1036  2                     .fao_strings[3],arglist[3] ) ;
;  624          1037  2
;  625          1038  2 !
;  626          1039  2 ! The line number will always be output so get the line info.
;  627          1040  2 !
;  628          1041  2 Arglist[0] = .csr1[line_num] ;
;  629          1042  2 Str_number = 0 ;
;  630          1043  2
;  631          1044  2 !
;  632          1045  2 ! Via the KMS command, determine whether to output the memory
;  633          1046  2 ! extension and lun information.
;  634          1047  2 !
;  635          1048  2 Case .csr1[cmd] from 0 to %x'B' of
;  636          1049  2     SET
;  637          1050  2     [0]:
;  638          1051  2         If .flag AND .csr3[cmd_sts] NEQ %x'36' OR
;  639          1052  2                      .csr3[cmd_sts] NEQ %x'35'
;  640          1053  2         Then
;  641          1054  3             Begin
;  642          1055  3             Arglist[0] = %ASCID 'LINE = N/A' ;
;  643          1056  3             Str_number = 3 ;
;  644          1057  2             End ;
;  645          1058  2
;  646          1059  2     [5,6]:
;  647          1060  3         Begin
;  648          1061  3         If .flag
;  649          1062  3         Then
;  650          1063  4             Begin
;  651          1064  4             Arglist[0] = .csr1[mem_ext] ;
;  652          1065  4             Arglist[1] = .csr1[line_num] ;
```

```
;  653          1066  4              Arglist[2] = .csr1[lun] ;
;  654          1067  4              Str_number = 1 ;
;  655          1068  4              End
;  656          1069  3          Else
;  657          1070  4              Begin
;  658          1071  4              If .csr1[cmd] EQL 5
;  659          1072  4              then
;  660          1073  5                  Begin
;  661          1074  5                  Arglist[1] = .csr1[lun] ;
;  662          1075  5                  Str_number = 2 ;
;  663          1076  4                  End ;
;  664          1077  3              End ;
;  665          1078  2          End ;
;  666          1079  2
;  667          1080  2      [%x'A']:
;  668          1081  3          Begin
;  669          1082  3          If .flag
;  670          1083  3          Then
;  671          1084  4              Begin
;  672          1085  4              Arglist[0] = .csr1[mem_ext] ;
;  673          1086  4              Arglist[1] = .csr1[line_num] ;
;  674          1087  4              Str_number = 10 ;
;  675          1088  3              End ;
;  676          1089  2          End ;
;  677          1090  2
;  678          1091  2      [%x'B'] :
;  679          1092  3          Begin
;  680          1093  3          Arglist[0] = .csr1[line_num] ;
;  681          1094  3          Arglist[1] = .csr1[lun] ;
;  682          1095  3          Str_number = 2 ;
;  683          1096  2          End ;
;  684          1097  2
;  685          1098  2      [Inrange]:
;  686          1099  3          Begin
;  687          1100  3          Arglist[0] = .csr1[line_num] ;
;  688          1101  3          Str_number = 0 ;
;  689          1102  2          End ;
;  690          1103  2      Tes ;
;  691          1104  2
;  692          1105  2 !
;  693          1106  2 ! Output the line, memory extension and lun information for CSR1.
;  694          1107  2 !
;  695          1108  2 OUTPUT_LINES ( .fao_strings[.str_number],arglist[0] ) ;
;  696          1109  2
;  697          1110  2
;  698          1111  2 !
;  699          1112  2 ! CSR2
;  700          1113  2 !
;  701          1114  2 ! Arglist contains: the register mnemonic, the starting position of the
;  702          1115  2 ! register contents field, the register contents, and the memory address.
;  703          1116  2 !
;  704          1117  2 Arglist[0] = CSTRING ('CSR2') ;
;  705          1118  2 Arglist[1] = (16 - .(.arglist[0])<0,8>)+4 ;
;  706          1119  2 Arglist[2] = ..csr2 ;
;  707          1120  2 Arglist[4] = .csr2[mem_addrs] ;
;  708          1121  2
;  709          1122  2 !
```

```
;   710          1123  2 ! Determine whether the memory address string or the mcu address
;   711          1124  2 ! string should be output. Get the address of the appropriate string.
;   712          1125  2 !
;   713          1126  2 Str_number = 4 ;
;   714          1127  2
;   715          1128  2 Case .csr1[cmd] from 0 to %x'B' of
;   716          1129  2     Set
;   717          1130  2     [0]:
;   718          1131  2         Arglist[3] = .mcu_string ;
;   719          1132  2     [1]:
;   720          1133  2         If .flag
;   721          1134  2         Then Arglist[3] = .mcu_string ;
;   722          1135  2
;   723          1136  2     [3,5,6,9,%x'A']:
;   724          1137  3         Begin
;   725          1138  3         Arglist[3] = .mem_string ;
;   726          1139  2         End ;
;   727          1140  2
;   728          1141  2     [%x'B']:
;   729          1142  3         Begin
;   730          1143  3         Arglist[3] = .csr2[sts_byte_0] ;
;   731          1144  3         Arglist[4] = .csr2[sts_byte_1] ;
;   732          1145  3         Str_number = 9 ;
;   733          1146  2         End ;
;   734          1147  2
;   735          1148  2     [Inrange]:
;   736          1149  3         Begin
;   737          1150  3         Str_number = -1 ;
;   738          1151  2         End ;
;   739          1152  2     Tes ;
;   740          1153  2
;   741          1154  2 !
;   742          1155  2 ! Determine whether to output the memory address bit to text information.
;   743          1156  2 !
;   744          1157  2 If .str_number EQL -1
;   745          1158  2 Then
;   746          1159  2     OUTPUT_LINES ( .fao_string1,arglist[0])
;   747          1160  2 else
;   748          1161  2     OUTPUT_LINES ( .fao_string1,arglist[0],
;   749          1162  2                    .fao_strings[.str_number],arglist[3] ) ;
;   750          1163  2
;   751          1164  2 !
;   752          1165  2 ! CSR3
;   753          1166  2 !
;   754          1167  2 ! Arglist contains: the register mnemonic, the starting position of the
;   755          1168  2 ! register contents field, and  the register contents.
;   756          1169  2 !
;   757          1170  2 Arglist[0] = CSTRING ('CSR3') ;
;   758          1171  2 Arglist[1] = (16 - .(.arglist[0])<0,8>)+4 ;
;   759          1172  2 Arglist[2] = ..csr3 ;
;   760          1173  2
;   761          1174  2 !
;   762          1175  2 ! Output the first part of the CSR3 information.
;   763          1176  2 !
;   764          1177  2 OUTPUT_LINES ( .fao_string1,arglist[0] ) ;
;   765          1178  2
;   766          1179  2 !
```

```
;   767          1180  2 ! Determine whether translating the output csr's.
;   768          1181  2 !
;   769          1182  2 If .flag
;   770          1183  2 Then
;   771          1184  2     !
;   772          1185  2     ! Yes, get the command status text.
;   773          1186  3     Begin
;   774          1187  3     Arglist[3] = reg_desc ;
;   775          1188  3     Str_number = 3 ;
;   776          1189  3
;   777          1190  3     Index = -1 ;
;   778          1191  3     If .csr3[cmd_sts] EQL 9
;   779          1192  3     Then Index = 0 ;
;   780          1193  3
;   781          1194  3     Case .csr3[cmd_sts] from %X'28' to %x'3F' of
;   782          1195  3         Set
;   783          1196  3         [%x'28']: Index = 1 ;
;   784          1197  3         [%x'2D']: Index = 2 ;
;   785          1198  3         [%x'2E']: Index = 3 ;
;   786          1199  3         [%x'2F']: Index = 4 ;
;   787          1200  3         [%x'30']: Index = 5 ;
;   788          1201  3         [%x'31']: Index = 6 ;
;   789          1202  3         [%x'32']: Index = 7 ;
;   790          1203  3         [%x'33']: Index = 8 ;
;   791          1204  3         [%x'34']: Index = 9 ;
;   792          1205  3         [%x'35']: Index = 10 ;
;   793          1206  3         [%x'36']: Index = 11 ;
;   794          1207  3         [%x'37']: Index = 12 ;
;   795          1208  3         [%x'38']: Index = 13 ;
;   796          1209  3         [%x'39']:
;   797          1210  4             Begin
;   798          1211  4             Index = -2 ;
;   799          1212  4             Case .csr1[cmd] from 0 to %x'B' of
;   800          1213  4                 Set
;   801          1214  4                 [0]:
;   802          1215  4                     Arglist[4] = $DESCRIPTOR ('MODEM ALREADY INITIALIZED') ;
;   803          1216  4
;   804          1217  4                 [1]:
;   805          1218  4                     Arglist[4] = $DESCRIPTOR ('MODEM NOT INITIALIZED OR LINE!/!39< !>ALREADY INITIALIZED') ;
;   806          1219  4
;   807          1220  4                 [2]:
;   808          1221  4                     Arglist[4] = $DESCRIPTOR ('LINE NOT INITIALIZED OR!/!39< !>NOT DISCONNECTED') ;
;   809          1222  4
;   810          1223  4                 [3,4]:
;   811          1224  4                     Arglist[4] = $DESCRIPTOR ('LINE NOT INITIALIZED OR!/!39< !>ALREADY CONNECTED') ;
;   812          1225  4
;   813          1226  4                 [5,6,%X'A',%X'B']:
;   814          1227  4                     Arglist[4] = $DESCRIPTOR ('LINE NOT INITIALIZED OR!/!39< !>NOT CONNECTED') ;
;   815          1228  4
;   816          1229  4                 [INRANGE]:
;   817          1230  5                     Begin
;   818          1231  5                     Arglist[4] = .unknown_desc ;
;   819          1232  5                     Arglist[5] = .csr3[cmd_sts] ;
;   820          1233  4                     End ;
;   821          1234  4                 Tes ;
;   822          1235  3             End ;
;   823          1236  3         [%x'3A']: Index = 14 ;
```

```
;  824          1237  3          [%x'3D']: Index = 15 ;
;  825          1238  3          [%x'3E']: Index = 16 ;
;  826          1239  3          [%x'3F']:
;  827          1240  4              Begin
;  828          1241  4              Index = 17 ;
;  829          1242  4              Case .csr1[cmd] from 1 to %x'A' of
;  830          1243  4                  Set
;  831          1244  4                  [1]:
;  832          1245  4                      Arglist[5] = CSTRING ('PARAMETER') ;
;  833          1246  4
;  834          1247  4                  [5,%X'A']:
;  835          1248  4                      Arglist[5] = CSTRING ('TRANSMIT') ;
;  836          1249  4
;  837          1250  4                  [6]:
;  838          1251  4                      Arglist[5] = CSTRING ('RECEIVE') ;
;  839          1252  4
;  840          1253  4                  [9]:
;  841          1254  4                      Arglist[5] = CSTRING ('STATISTICS MEM') ;
;  842          1255  4
;  843          1256  4                  [INRANGE]:
;  844          1257  5                      Begin
;  845          1258  5                      Arglist[4] = .unknown_desc ;
;  846          1259  5                      Arglist[5] = .csr3[cmd_sts] ;
;  847          1260  4                      End ;
;  848          1261  4                  Tes ;
;  849          1262  3              End ;
;  850          1263  3
;  851          1264  3          [INRANGE]:
;  852          1265  3              Index = -1
;  853          1266  3              Tes ;
;  854          1267  3
;  855          1268  3          !
;  856          1269  3          ! Determine whether the command status code was found.
;  857          1270  3          !
;  858          1271  3          If .index GTR 0
;  859          1272  3          Then
;  860          1273  3              !
;  861          1274  3              ! Yes, get the address of the descriptor.
;  862          1275  3              !
;  863          1276  3              Arglist[4] = cmdsts_desc_tbl[.index,0,0,0,0] ;
;  864          1277  3          !
;  865          1278  3          ! Determine whether the command status code was found.
;  866          1279  3          !
;  867          1280  3          If .index EQL -1
;  868          1281  3          Then
;  869          1282  3              !
;  870          1283  3              ! Determine if it was one of the completion msgs and
;  871          1284  3              ! build the appropriate msg.
;  872          1285  3              !
;  873          1286  4              Begin
;  874          1287  4              Arglist[6] = CSTRING ('FROM LUN') ;
;  875          1288  4              Index2 = -1 ;
;  876          1289  4
;  877          1290  4              Case .csr3[cmd_sts] from 1 to %x'17' of
;  878          1291  4                  Set
;  879          1292  4                  [1]: Index2 = 2 ;
;  880          1293  4                  [3]: Index2 = 0 ;
```

```
;  881          1294  4              [5]: Index2 = 1 ;
;  882          1295  4              [7]: Index2 = 3 ;
;  883          1296  4              [%x'11']:
;  884          1297  5                  Begin
;  885          1298  5                  Index2 = 2 ;
;  886          1299  5                  Arglist[6] = CSTRING ('OF TEST REQUEST MSG FROM LINE') ;
;  887          1300  4                  End ;
;  888          1301  4
;  889          1302  4              [%x'13']:
;  890          1303  5                  Begin
;  891          1304  5                  Index2 = 0 ;
;  892          1305  5                  Arglist[6] = CSTRING ('OF TEST REQUEST MSG FROM LINE') ;
;  893          1306  4                  End ;
;  894          1307  4
;  895          1308  4              [%x'15']:
;  896          1309  5                  Begin
;  897          1310  5                  Index2 = 1 ;
;  898          1311  5                  Arglist[6] = CSTRING ('OF TEST REQUEST MSG FROM LINE') ;
;  899          1312  4                  End ;
;  900          1313  4
;  901          1314  4              [%x'17']:
;  902          1315  5                  Begin
;  903          1316  5                  Index2 = 3 ;
;  904          1317  5                  Arglist[6] = CSTRING ('OF TEST REQUEST MSG FROM LINE') ;
;  905          1318  4                  End ;
;  906          1319  4
;  907          1320  4              [INRANGE]:
;  908          1321  4                  Index2 = -1 ;
;  909          1322  4              Tes ;
;  910          1323  4
;  911          1324  4          !
;  912          1325  4          ! Determine if it was one of the completion msgs.
;  913          1326  4          !
;  914          1327  4          If .index2 NEQ -1
;  915          1328  4          Then
;  916          1329  4              !
;  917          1330  4              ! Yes, get the fao control string and the msg.
;  918          1331  4              !
;  919          1332  5              Begin
;  920          1333  5              Arglist[4] = .fao_strings[12] ;
;  921          1334  5              Arglist[5] = .msg[.index2] ;
;  922          1335  5              End
;  923          1336  4          Else
;  924          1337  4              !
;  925          1338  4              ! No, get the address of the unknown msg descriptor and
;  926          1339  4              ! the command sts code.
;  927          1340  4              !
;  928          1341  5              Begin
;  929          1342  5              Arglist[4] = .unknown_desc ;
;  930          1343  5              Arglist[5] = .csr3[cmd_sts] ;
;  931          1344  4              End ;
;  932          1345  3          End ;
;  933          1346  3
;  934          1347  3      !
;  935          1348  3      ! Format the string.
;  936          1349  3      !
;  937          1350  3      Reg_desc[dsc$w_length] = %ALLOCATION(reg_buffer) ;
```

```
;  938          1351  3     FORMAT_STRING (arglist[4],reg_desc,arglist[5]) ;
;  939          1352  2     End ;
;  940          1353  2
;  941          1354  2 !
;  942          1355  2 ! Determine whether to output the byte count and memory extension information.
;  943          1356  2 !
;  944          1357  2 Case .csr1[cmd]from 0 to %x'B' of
;  945          1358  2     Set
;  946          1359  2     [0,1,3,%x'9']:
;  947          1360  3         Begin
;  948          1361  3         If NOT .flag
;  949          1362  3         Then
;  950          1363  3             !
;  951          1364  3             ! Translating input csr's, get the input memory extension info.
;  952          1365  3             !
;  953          1366  4             Begin
;  954          1367  4             Arglist[3] = .csr3[inp_mem_ext] ;
;  955          1368  4             Str_number = 8 ;
;  956          1369  4             End
;  957          1370  3         Else
;  958          1371  3             !
;  959          1372  3             ! Translating output csr's, get the output memory extension info.
;  960          1373  3             !
;  961          1374  4             Begin
;  962          1375  4             Arglist[3] = .csr3[out_mem_ext] ;
;  963          1376  4             Arglist[4] = reg_desc ;
;  964          1377  4             Str_number = 5 ;
;  965          1378  3             End ;
;  966          1379  2         End ;
;  967          1380  2
;  968          1381  2     [5,6,%x'A']:
;  969          1382  3         Begin
;  970          1383  3         If NOT .flag
;  971          1384  3         Then
;  972          1385  3             !
;  973          1386  3             ! Translating input csr's, get the byte count and the
;  974          1387  3             ! input memory extension info.
;  975          1388  3             !
;  976          1389  4             Begin
;  977          1390  4             Arglist[3] = .csr3[byte_cnt] ;
;  978          1391  4             Arglist[4] = .csr3[inp_mem_ext] ;
;  979          1392  4             Str_number = 6 ;
;  980          1393  4             End
;  981          1394  3         Else
;  982          1395  3             !
;  983          1396  3             ! Translating output csr's, get the byte count info.
;  984          1397  3             !
;  985          1398  4             Begin
;  986          1399  4             Arglist[3] = .csr3[byte_cnt] ;
;  987          1400  4             Arglist[4] = reg_desc ;
;  988          1401  4             Str_number = 7 ;
;  989          1402  3             End ;
;  990          1403  2         End ;
;  991          1404  2
;  992          1405  2     [Inrange]:
;  993          1406  3         Begin
;  994          1407  3         If NOT .flag
```

```
;  995          1408  3          Then
;  996          1409  3               !
;  997          1410  3               ! Translating input csr's, nothing left to output, return
;  998          1411  3               ! to the calling routine.
;  999          1412  3               !
; 1000          1413  3               Return
; 1001          1414  3          Else
; 1002          1415  3               !
; 1003          1416  3               ! Translating output csr's, set up the fao control string
; 1004          1417  3               ! number for the output of the command sts.
; 1005          1418  3               !
; 1006          1419  3               Str_number = 3 ;
; 1007          1420  2          End ;
; 1008          1421  2      Tes ;
; 1009          1422  2
; 1010          1423  2 !
; 1011          1424  2 ! Output the remainder of the csr3 information.
; 1012          1425  2 !
; 1013          1426  2 OUTPUT_LINES ( .fao_strings[.str_number],arglist[3] ) ;
; 1014          1427  2
; 1015          1428  1 End ;            ! Routine


                                                                .PSECT  $PLIT,NOWRT,NOEXE,  PIC,2

                                                        0105A           .BLKB   2
          00  53  41  21  3E  21  20  3C  31  21  2F  21  0105C P.AEZ:   .ASCII  \!/!1< !>!AS\<0>                                ;
                                                    010E000B  01068 P.AEY:   .LONG   17694731                                   ;
                                                    00000000' 0106C          .ADDRESS P.AEZ                                     ;
                                                30  52  53  43  04  01070 P.AFA:   .ASCII  <4>\CSR0\                           ;
                                                31  52  53  43  04  01075 P.AFB:   .ASCII  <4>\CSR1\                           ;
                                                                    0107A        .BLKB   2
                      00  00  41  2F  4E  20  3D  20  45  4E  49  4C  0107C P.AFD:   .ASCI'  \LINE = N/A\<0><0>                 ;
                                                    010E000A  01088 P.AFC:   .LONG   17694730                                   ;
                                                    00000000' 0108C          .ADDRESS P.AFD                                     ;
                                                32  52  53  43  04  01090 P.AFE:   .ASCII  <4>\CSR2\                           ;
                                                33  52  53  43  04  01095 P.AFF:   .ASCII  <4>\CSR3\                           ;
49  20  59  44  41  45  52  4C  41  20  4D  45  44  4F  4D  0109A P.AFH:   .ASCII  \MODEM ALREADY INITIALIZED\                    ;
                          44  45  5A  49  4C  41  49  54  49  4E  010A9                                                         ;
                                                                    010B3        .BLKB   1                                      ;
                                                    00000019  010B4 P.AFG:   .LONG   25                                         ;
                                                    00000000' 010B8          .ADDRESS P.AFH                                     ;
49  54  49  4E  49  20  54  4F  4E  20  4D  45  44  4F  4D  010BC P.AFJ:   .ASCII  \MODEM NOT INITIALIZED OR LINE!/!39< !>AL\    ;
21  45  4E  49  4C  20  52  4F  20  44  45  5A  49  4C  41  010CB                                                               ;
                          4C  41  3E  21  20  3C  39  33  21  2F  010DA                                                         ;
5A  49  4C  41  49  54  49  4E  49  20  59  44  41  45  52  010E4        .ASCII  \READY INITIALIZED\                            ;
                                                44  45      010F3                                                               ;
                                                                    010F5        .BLKB   3                                      ;
                                                    00000039  010F8 P.AFI:   .LONG   57                                         ;
                                                    00000000' 010FC          .ADDRESS P.AFJ                                     ;
41  49  54  49  4E  49  20  54  4F  4E  20  45  4E  49  4C  01100 P.AFL:   .ASCII  \LINE NOT INITIALIZED OR!/!39< !>NOT DISC\    ;
20  3C  39  33  21  2F  21  52  4F  20  44  45  5A  49  4C  0110F                                                               ;
                          43  53  49  44  20  54  4F  4E  3E  21  0111E                                                         ;
                                      44  45  54  43  45  4E  4E  4F  01128        .ASCII  \ONNECTED\                         ;
                                                    00000030  01130 P.AFK:   .LONG   48                                         ;
                                                    00000000' 01134          .ADDRESS P.AFL                                     ;
```

```
41 49 54 49 4E 49 20 54 4F 4E 20 45 4E 49 4C 01138 P.AFN:  .ASCII  \LINE NOT INITIALIZED OR!/!39< !>ALREADY \
20 3C 39 33 21 2F 21 52 4F 20 44 45 5A 49 4C 01147
                20 59 44 41 45 52 4C 41 3E 21 01156
                   44 45 54 43 45 4E 4E 4F 43 01160          .ASCII  \CONNECTED\
                                             01169          .BLKB   3
                                  00000031   0116C P.AFM:   .LONG   49
                                  00000000'  01170          .ADDRESS P.AFN
41 49 54 49 4E 49 20 54 4F 4E 20 45 4E 49 4C 01174 P.AFP:  .ASCII  \LINE NOT INITIALIZED OR!/!39< !>NOT CONN\
20 3C 39 33 21 2F 21 52 4F 20 44 45 5A 49 4C 01183
                4E 4E 4F 43 20 54 4F 4E 3E 21 01192
                            44 45 54 43 45 01199          .ASCII  \ECTED\
                                             011A1          .BLKB   3
                                  0000002D   011A4 P.AFO:   .LONG   45
                                  00000000'  011A8          .ADDRESS P.AFP
                52 45 54 45 4D 41 52 41 50 09 011AC P.AFQ:  .ASCII  <9>\PARAMETER\
                   54 49 4D 53 4E 41 52 54 08 011B6 P.AFR:  .ASCII  <8>\TRANSMIT\
                      45 56 49 45 43 45 52 07 011BF P.AFS:  .ASCII  <7>\RECEIVE\
4D 45 4D 20 53 43 49 54 53 49 54 41 54 53 0E 011C7 P.AFT:  .ASCII  <14>\STATISTICS MEM\
                   4E 55 4C 20 4D 4F 52 46 08 011D6 P.AFU:  .ASCII  <8>\FROM LUN\
53 45 55 51 45 52 20 54 53 45 54 20 46 4F 1D 011DF P.AFV:  .ASCII  <29>\OF TEST REQUEST MSG FROM LINE\
45 4E 49 4C 20 4D 4F 52 46 20 47 53 4D 20 54 011EE
53 45 55 51 45 52 20 54 53 45 54 20 46 4F 1D 011FD P.AFW:  .ASCII  <29>\OF TEST REQUEST MSG FROM LINE\
45 4E 49 4C 20 4D 4F 52 46 20 47 53 4D 20 54 0120C
53 45 55 51 45 52 20 54 53 45 54 20 46 4F 1D 0121B P.AFX:  .ASCII  <29>\OF TEST REQUEST MSG FROM LINE\
45 4E 49 4C 20 4D 4F 52 46 20 47 53 4D 20 54 0122A
53 45 55 51 45 52 20 54 53 45 54 20 46 4F 1D 01239 P.AFY:  .ASCII  <29>\OF TEST REQUEST MSG FROM LINE\
45 4E 49 4C 20 4D 4F 52 46 20 47 53 4D 20 54 01248

                                             01248          .PSECT  $OWN$,NOEXE,  PIC,2

                                  00338 CSR0:    .BLKB   4
                                  0033C CSR1:    .BLKB   4
                                  00340 CSR2:    .BLKB   4
                                  00344 CSR3:    .BLKB   4


                                             00000          .PSECT  $CODE,NOWRT,  PIC,2

                                  0FFC 00000 OUTPUT_REGISTERS:
                                             00000          .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 0895
      5B 00000000G   00  9E 00002          00002          MOVAB   TRANSLATE_BITS, R11                           ;
      5A 00000000G   00  9E 00009          00009          MOVAB   REGSAV+8, R10                                 ;
      59 00000000G   00  9E 00010          00010          MOVAB   OUTPUT_LINES, R9                              ;
      58 00000000'   00  9E 00017          00017          MOVAB   P.AEY, R8                                     ;
      57 00000000'   00  9E 0001E          0001E          MOVAB   ARGLIST, R7                                   ;
      5E     FDF4    CE  9E 00025          00025          MOVAB   -524(SP), SP                                  ;
04    AE 010E0000    8F  D0 0002A          0002A          MOVL    #17694720, REG_DESC                           ; 0929
08    AE       0C    AE  9E 00032          00032          MOVAB   REG_BUFFER, REG_DESC+4                        ;
      54       04    AC  D0 00037          00037          MOVL    FLAG, R4                                      ; 0941
      1E             54  E8 0003B          0003B          BLBS    R4, 1$                                        ;
0334  C7             6A  9E 0003E          0003E          MOVAB   REGSAV+8, CSR0                                ; 0947
0338  C7       04    AA  9E 00043          00043          MOVAB   REGSAV+12, CSR1                               ; 0948
033C  C7       08    AA  9E 00049          00049          MOVAB   REGSAV+16, CSR2                               ; 0949
0340  C7       0C    AA  9E 0004F          0004F          MOVAB   REGSAV+20, CSR3                               ; 0950
      67 00A0        C7  D0 00055          00055          MOVL    INPUT_HEADER, ARGLIST                         ; 0952
                     1D  11 0005A          0005A          BRB     2$                                            ; 0941
```

```
                     0334  C7    10  AA  9E 0005C 1$:    MOVAB   REGSAV+24, CSR0                           ; 0959
                     0338  C7    14  AA  9E 00062        MOVAB   REGSAV+28, CSR1                           ; 0960
                     033C  C7    18  AA  9E 00068        MOVAB   REGSAV+32, CSR2                           ; 0961
                     0340  C7    1C  AA  9E 0006E        MOVAB   REGSAV+36, CSR3                           ; 0962
                           67  00C0  C7  D0 00074        MOVL    OUTPUT_HEADER, ARGLIST                    ; 0964
                                     57  DD 00079 2$:    PUSHL   R7                                        ; 0970
                                     58  DD 0007B        PUSHL   R8                                        ;
                           69        02  FB 0007D        CALLS   #2, OUTPUT_LINES                          ;
                           67    08  A8  9E 00080        MOVAB   P.AFA, ARGLIST                            ; 0980
                           50        67  D0 00084        MOVL    ARGLIST, R0                               ; 0981
                     04    A7        60  9A 00087        MOVZBL  (R0), ARGLIST+4                           ;
          04   A7          14    04  A7  C3 0008B        SUBL3   ARGLIST+4, #20, ARGLIST+4                 ;
                           50  0334  C7  D0 00091        MOVL    CSR0, R0                                  ; 0982
                     08    A7        60  D0 00096        MOVL    (R0), ARGLIST+8                           ;
                                     6E  D4 0009A        CLRL    (SP)                                      ; 0988
                                     5E  DD 0009C        PUSHL   SP                                        ;
                                 64  A7  9F 0009E        PUSHAB  FAO_STRING                                ; 0987
                               00BC  C7  9F 000A1        PUSHAB  OUT_ARGLIST                               ;
                               02DC  C7  9F 000A5        PUSHAB  CSR0_DESC_TBL                             ;
                                 58  A7  9F 000A9        PUSHAB  CSR0_MASK                                 ;
                                     50  DD 000AC        PUSHL   R0                                        ;
                           6B        06  FB 000AE        CALLS   #6, TRANSLATE_BITS                        ;
                     00C4  C7        50  D0 000B1        MOVL    R0, STATUS                                ;
                           11        50  E9 000B6        BLBC    R0, 3$                                    ;
                               00BC  C7  DD 000B9        PUSHL   OUT_ARGLIST                               ; 0995
                                 64  A7  DD 000BD        PUSHL   FAO_STRING                                ;
                                     57  DD 000C0        PUSHL   R7                                        ; 0994
                                 68  A7  DD 000C2        PUSHL   FAO_STRING1                               ; 0995
                           69        04  FB 000C5        CALLS   #4, OUTPUT_LINES                          ;
                                     08  11 000C8        BRB     4$                                        ; 0987
                                     57  DD 000CA 3$:    PUSHL   R7                                        ; 1001
                                 68  A7  DD 000CC        PUSHL   FAO_STRING1                               ;
                           69        02  FB 000CF        CALLS   #2, OUTPUT_LINES                          ;
                           67    0D  A8  9E 000D2 4$:    MOVAB   P.AFB, ARGLIST                            ; 1012
                           50        67  D0 000D6        MOVL    ARGLIST, R0                               ; 1013
                     04    A7        60  9A 000D9        MOVZBL  (R0), ARGLIST+4                           ;
          04   A7          14    04  A7  C3 000DD        SUBL3   ARGLIST+4, #20, ARGLIST+4                 ;
                           51  0338  C7  D0 000E3        MOVL    CSR1, R1                                  ; 1014
                     08    A7        61  D0 000E8        MOVL    (R1), ARGLIST+8                           ;
   50          61          04        00  EF 000EC        EXTZV   #0, #4, (R1), R0                          ; 1015
                     0C    A7  00CC C740 7E 000F1        MOVAQ   KMSCMDS_DESC_TBL[R0], ARGLIST+12          ;
                                     6E  D4 000F8        CLRL    (SP)                                      ; 1021
                                     5E  DD 000FA        PUSHL   SP                                        ;
                                 64  A7  9F 000FC        PUSHAB  FAO_STRING                                ; 1020
                               00BC  C7  9F 000FF        PUSHAB  OUT_ARGLIST                               ;
                               0324  C7  9F 00103        PUSHAB  CSR1_DESC_TBL                             ;
                                 5C  A7  9F 00107        PUSHAB  CSR1_MASK                                 ;
                                     51  DD 0010A        PUSHL   R1                                        ;
                           6B        06  FB 0010C        CALLS   #6, TRANSLATE_BITS                        ;
                     00C4  C7        50  D0 0010F        MOVL    R0, STATUS                                ;
                           17        50  E9 00114        BLBC    R0, 5$                                    ;
                               00BC  C7  DD 00117        PUSHL   OUT_ARGLIST                               ; 1029
                                 64  A7  DD 0011B        PUSHL   FAO_STRING                                ;
                                 0C  A7  9F 0011E        PUSHAB  ARGLIST+12                                ; 1028
                                 78  A7  DD 00121        PUSHL   FAO_STRINGS+12                            ; 1029
                                     57  DD 00124        PUSHL   R7                                        ; 1027
                                 68  A7  DD 00126        PUSHL   FAO_STRING1                               ; 1029
```

```
                                 69           06  FB 00129        CALLS   #6, OUTPUT_LINES                       :
                                              0E  11 0012C        BRB     6$                                     : 1020
                                        0C    A7  9F 0012E 5$:    PUSHAB  ARGLIST+12                             : 1036
                                        78    A7  DD 00131        PUSHL   FAO_STRINGS+12                         :
                                              57  DD 00134        PUSHL   R7                                     : 1035
                                        68    A7  DD 00136        PUSHL   FAO_STRING1                            :
                                 69           04  FB 00139        CALLS   #4, OUTPUT_LINES                       :
                                 50     0338  C7  DO 0013C 6$:    MOVL    CSR1, R0                               : 1041
       67        01      A0      02           00  EF 00141        EXTZV   #0, #2, 1(R0), ARGLIST                 :
                                              52  D4 00147        CLRL    STR_NUMBER                             : 1042
       53                60      04           00  EF 00149        EXTZV   #0, #4, (R0), R3                       : 1048
                         0B      00           53  CF 0014E        CASEL   R3, #0, #11                            :
     0084              0084    0084         0018     00152 7$:    .WORD   8$-7$,-                                :
     0084              003C    003C         0084     0015A                16$-7$,-                               :
     0073              005F    0084         0084     00162                16$-7$,-                               :
                                                                          16$-7$,-                               :
                                                                          16$-7$,-                               :
                                                                          11$-7$,-                               :
                                                                          11$-7$,-                               :
                                                                          16$-7$,-                               :
                                                                          16$-7$,-                               :
                                                                          16$-7$,-                               :
                                                                          13$-7$,-                               :
                                                                          14$-7$                                 :
                                 0C           54  E9 0016A 8$:    BLBC    R4, 9$                                 : 1051
                                 51     0340  C7  D0 0016D        MOVL    CSR3, R1                               :
       36                61      06           0A  ED 00172        CMPZV   #10, #6, (R1), #54                     :
                                              0C  12 00177        BNEQ    10$                                    :
                                 51     0340  C7  D0 00179 9$:    MOVL    CSR3, R1                               : 1052
       35                61      06           0A  ED 0017E        CMPZV   #10, #6, (R1), #53                     :
                                              59  13 00183        BEQL    17$                                    :
                                 67       20  A8  9E 00185 10$:   MOVAB   P.AFC, ARGLIST                         : 1055
                                 52           03  D0 00189        MOVL    #3, STR_NUMBER                         : 1056
                                 50           11  11 0018C        BRB     17$                                    : 1051
                                 17           54  E9 0018E 11$:   BLBC    R4, 12$                                : 1061
       67                60      02           05  EF 00191        EXTZV   #5, #2, (R0), ARGLIST                  : 1064
  04   A7        01      A0      02           00  EF 00196        EXTZV   #0, #2, 1(R0), ARGLIST+4               : 1065
  08   A7                60      05           0B  EF 0019D        EXTZV   #11, #5, (R0), ARGLIST+8               : 1066
                                 52           01  D0 001A3        MOVL    #1, STR_NUMBER                         : 1067
                                              36  11 001A6        BRB     17$                                    : 1061
       05                60      04           00  ED 001A8 12$:   CMPZV   #0, #4, (R0), #5                       : 1071
                                              2F  12 001AD        BNEQ    17$                                    :
                                              1A  11 001AF        BRB     15$                                    : 1074
                                 2A           54  E9 001B1 13$:   BLBC    R4, 17$                                : 1082
       67                60      02           05  EF 001B4        EXTZV   #5, #2, (R0), ARGLIST                  : 1085
  04   A7        01      A0      02           00  EF 001B9        EXTZV   #0, #2, 1(R0), ARGLIST+4               : 1086
                                 52           0A  D0 001C0        MOVL    #10, STR_NUMBER                        : 1087
                                              19  11 001C3        BRB     17$                                    : 1048
       67        01      A0      02           00  EF 001C5 14$:   EXTZV   #0, #2, 1(R0), ARGLIST                 : 1093
  04   A7                60      05           0B  EF 001CB 15$:   EXTZV   #11, #5, (R0), ARGLIST+4               : 1094
                                 52           02  D0 001D1        MOVL    #2, STR_NUMBER                         : 1095
                                              08  11 001D4        BRB     17$                                    : 1048
       67        01      A0      02           00  EF 001D6 16$:   EXTZV   #0, #2, 1(R0), ARGLIST                 : 1100
                                              52  D4 001DC        CLRL    STR_NUMBER                             : 1101
                                              57  DD 001DE 17$:   PUSHL   R7                                     : 1108
                                      6C A742     DD 001E0        PUSHL   FAO_STRINGS[STR_NUMBER]                :
                                 69           02  FB 001E4        CALLS   #2, OUTPUT_LINES                       :
```

```
                              67      28   A8  9E 001E7        MOVAB   P.AFE, ARGLIST                        ; 1117
                              50           67  D0 001EB        MOVL    ARGLIST, R0                           ; 1118
                         04   A7           60  9A 001EE        MOVZBL  (R0), ARGLIST+4                       ;
            04     A7         14      04   A7  C3 001F2        SUBL3   ARGLIST+4, #20, ARGLIST+4             ;
                              50    033C   C7  D0 001F8        MOVL    CSR2, R0                              ; 1119
                         08   A7           60  D0 001FD        MOVL    (R0), ARGLIST+8                       ;
                         10   A7           60  3C 00201        MOVZWL  (R0), ARGLIST+16                      ; 1120
                              52           04  D0 00205        MOVL    #4, STR_NUMBER                        ; 1126
                              51    0338   C7  D0 00208        MOVL    CSR1, R1                              ; 1128
     53            61         04           00  EF 0020D        EXTZV   #0, #4, (R1), R3                      ;
                   0B         00           53  CF 00212        CASEL   R3, #0, #11                           ;
   0025          003B       001A         001D     00216 18$:   .WORD   20$-18$,-                             ;
   003B          0025       0025         003B     0021E                19$-18$,-                             ;
   002D          0025       0025         003B     00226                23$-18$,-                             ;
                                                                       21$-18$,-                             ;
                                                                       23$-18$,-                             ;
                                                                       21$-18$,-                             ;
                                                                       21$-18$,-                             ;
                                                                       23$-18$,-                             ;
                                                                       23$-18$,-                             ;
                                                                       21$-18$,-                             ;
                                                                       21$-18$,-                             ;
                                                                       22$-18$                               ;
                                           03  11 0022E        BRB     20$                                   ; 1131
                              21           54  E9 00230 19$:   BLBC    R4, 24$                               ; 1133
                         0C   A7    00A4   C7  D0 00233 20$:   MOVL    MCU_STRING, ARGLIST+12                ; 1134
                                           19  11 00239        BRB     24$                                   ; 1133
                         0C   A7    00A8   C7  D0 0023B 21$:   MOVL    MEM_STRING, ARGLIST+12                ; 1138
                                           11  11 00241        BRB     24$                                   ; 1128
                         0C   A7           60  9A 00243 22$:   MOVZBL  (R0), ARGLIST+12                      ; 1143
                         10   A7      01   A0  9A 00247        MOVZBL  1(R0), ARGLIST+16                     ; 1144
                              52           09  D0 0024C        MOVL    #9, STR_NUMBER                        ; 1145
                                           03  11 0024F        BRB     24$                                   ; 1128
                              52           01  CE 00251 23$:   MNEGL   #1, STR_NUMBER                        ; 1150
                              53      68   A7  D0 00254 24$:   MOVL    FAO_STRING1, R3                       ; 1159
                FFFFFFFF      8F           52  D1 00258        CMPL    STR_NUMBER, #-1                       ; 1157
                                           09  12 0025F        BNEQ    25$                                   ;
                                    0088   8F  BB 00261        PUSHR   #^M<R3,R7>                            ; 1159
                              69           02  FB 00265        CALLS   #2, OUTPUT_LINES                      ;
                                           0E  11 00268        BRB     26$                                   ;
                                      0C   A7  9F 0026A 25$:   PUSHAB  ARGLIST+12                            ; 1162
                                      6C A742  DD 0026D        PUSHL   FAO_STRINGS[STR_NUMBER]               ;
                                    0088   8F  BB 00271        PUSHR   #^M<R3,R7>                            ; 1161
                              69           04  FB 00275        CALLS   #4, OUTPUT_LINES                      ;
                              67      2D   A8  9E 00278 26$:   MOVAB   P.AFF, ARGLIST                        ; 1170
                              50           67  D0 0027C        MOVL    ARGLIST, R0                           ; 1171
                         04   A7           60  9A 0027F        MOVZBL  (R0), ARGLIST+4                       ;
            04     A7         14      04   A7  C3 00283        SUBL3   ARGLIST+4, #20, ARGLIST+4             ;
                              50    0340   C7  D0 00289        MOVL    CSR3, R0                              ; 1172
                         08   A7           60  D0 0028E        MOVL    (R0), ARGLIST+8                       ;
                                           57  DD 00292        PUSHL   R7                                    ; 1177
                                      68   A7  DD 00294        PUSHL   FAO_STRING1                           ;
                              69           02  FB 00297        CALLS   #2, OUTPUT_LINES                      ;
                              03           54  E8 0029A        BLBS    R4, 27$                               ; 1182
                                         0230  31 0029D        BRW     84$                                   ;
                         0C   A7      04   AE  9E 002A0 27$:   MOVAB   REG_DESC, ARGLIST+12                  ; 1187
                              52           03  D0 002A5        MOVL    #3, STR_NUMBER                        ; 1188
```

```
                                50              01  CE 002AB               MNEGL   #1, INDEX                                       ; 1190
                                51      0340    C7  D0 002AB               MOVL    CSR3, R1                                        ; 1191
            09                  61              06  ED 002B0               CMPZV   #10, #6, (R1), #9                               ;
                                                02  12 002B5               BNEQ    28$                                             ;
                                                50  D4 002B7               CLRL    INDEX                                           ; 1192
            55                  61              06  EF 002B9 28$:          EXTZV   #10, #6, (R1), R5                               ; 1194
                                17              28  CF 002BE               CASEL   R5, #40, #23                                    ;
          0123                0123            0123      0030     002C2 29$:      .WORD   30$-29$,-                                       ;
          003F                003A            0035      0123     002CA           67$-29$,-                                       ;
          0053                004E            0049      0044     002D2           67$-29$,-                                       ;
          0067                0062            005D      0058     002DA           67$-29$,-                                       ;
          0123                00C1            0071      006C     002E2           67$-29$,-                                       ;
          00D0                00CB            00C6      0123     002EA           31$-29$,-                                       ;
                                                                                   32$-29$,-                                       ;
                                                                                   33$-29$,-                                       ;
                                                                                   34$-29$,-                                       ;
                                                                                   35$-29$,-                                       ;
                                                                                   36$-29$,-                                       ;
                                                                                   37$-29$,-                                       ;
                                                                                   38$-29$,-                                       ;
                                                                                   39$-29$,-                                       ;
                                                                                   40$-29$,-                                       ;
                                                                                   41$-29$,-                                       ;
                                                                                   42$-29$,-                                       ;
                                                                                   43$-29$,-                                       ;
                                                                                   53$-29$,-                                       ;
                                                                                   67$-29$,-                                       ;
                                                                                   67$-29$,-                                       ;
                                                                                   55$-29$,-                                       ;
                                                                                   57$-29$,-                                       ;
                                                                                   59$-29$                                         ;
                50                              01  D0 002F2 30$:          MOVL    #1, INDEX                                       ; 1196
                                                7A  11 002F5               BRB     48$                                             ;
                50                              02  D0 002F7 31$:          MOVL    #2, INDEX                                       ; 1197
                                                7D  11 002FA               BRB     50$                                             ;
                50                              03  D0 002FC 32$:          MOVL    #3, INDEX                                       ; 1198
                                                78  11 002FF               BRB     50$                                             ;
                50                              04  D0 00301 33$:          MOVL    #4, INDEX                                       ; 1199
                                                7B  11 00304               BRB     52$                                             ;
                50                              05  D0 00306 34$:          MOVL    #5, INDEX                                       ; 1200
                                                7B  11 00309               BRB     54$                                             ;
                50                              06  D0 0030B 35$:          MOVL    #6, INDEX                                       ; 1201
                                                7B  11 0030E               BRB     56$                                             ;
                50                              07  D0 00310 36$:          MOVL    #7, INDEX                                       ; 1202
                                                7B  11 00313               BRB     58$                                             ;
                50                              08  D0 00315 37$:          MOVL    #8, INDEX                                       ; 1203
                                                76  11 00318               BRB     58$                                             ;
                50                              09  D0 0031A 38$:          MOVL    #9, INDEX                                       ; 1204
                                                71  11 0031D               BRB     58$                                             ;
                50                              0A  D0 0031F 39$:          MOVL    #10, INDEX                                      ; 1205
                                                6C  11 00322               BRB     58$                                             ;
                50                              0B  D0 00324 40$:          MOVL    #11, INDEX                                      ; 1206
                                                67  11 00327               BRB     58$                                             ;
                50                              0C  D0 00329 41$:          MOVL    #12, INDEX                                      ; 1207
                                                62  11 0032C               BRB     58$                                             ;
                50                              0D  D0 0032E 42$:          MOVL    #13, INDEX                                      ; 1208
                                                5D  11 00331               BRB     58$                                             ;
```

```
                          50          02  CE 00333 43$:    MNEGL   #2, INDEX                          ; 1211
                          53   0338   C7  D0 00336         MOVL    CSR1, R3                           ; 1212
      56          63      04          00  EF 0033B         EXTZV   #0, #4, (R3), R6                   ;
                  0B      00          56  CF 00340         CASEL   R6, #0, #11                        ;
    002F        0027    001F        0018     00344 44$:    .WORD   45$-44$,-                          ;
    0093        0037    0037        002F     0034C                 46$-44$,-                          ;
    0037        0037    0093        0093     00354                 47$-44$,-                          ;
                                                                   49$-44$,-                          ;
                                                                   49$-44$,-                          ;
                                                                   51$-44$,-                          ;
                                                                   51$-44$,-                          ;
                                                                   66$-44$,-                          ;
                                                                   66$-44$,-                          ;
                                                                   66$-44$,-                          ;
                                                                   51$-44$,-                          ;
                                                                   51$-44$                            ;
                      10  A7    4C    A8  9E 0035C 45$:    MOVAB   P.AFG, ARGLIST+16                  ; 1215
                                      72  11 00361         BRB     65$                                ;
                      10  A7  0090    C8  9E 00363 46$:    MOVAB   P.AFI, ARGLIST+16                  ; 1218
                                      7D  11 00369         BRB     68$                                ;
                      10  A7  00C8    C8  9E 0036B 47$:    MOVAB   P.AFK, ARGLIST+16                  ; 1221
                                      75  11 00371 48$:    BRB     68$                                ;
                      10  A7  0104    C8  9E 00373 49$:    MOVAB   P.AFM, ARGLIST+16                  ; 1224
                                      6D  11 00379 50$:    BRB     68$                                ;
                      10  A7  013C    C8  9E 0037B 51$:    MOVAB   P.AFO, ARGLIST+16                  ; 1227
                                      65  11 00381 52$:    BRB     68$                                ;
                          50          0E  D0 00383 53$:    MOVL    #14, INDEX                         ; 1236
                                      60  11 00386 54$:    BRB     68$                                ;
                          50          0F  D0 00388 55$:    MOVL    #15, INDEX                         ; 1237
                                      5B  11 0038B 56$:    BRB     68$                                ;
                          50          10  D0 0038D 57$:    MOVL    #16, INDEX                         ; 1238
                                      56  11 00390 58$:    BRB     68$                                ;
                          50          11  D0 00392 59$:    MOVL    #17, INDEX                         ; 1241
                          53   0338   C7  D0 00395         MOVL    CSR1, R3                           ; 1242
      56          63      04          00  EF 0039A         EXTZV   #0, #4, (R3), R6                   ;
                  09      01          56  CF 0039F         CASEL   R6, #1, #9                         ;
    0034        0034    0034        0014     003A3 60$:    .WORD   61$-60$,-                          ;
    0034        0034    0024        001C     003AB                 66$-60$,-                          ;
                        001C        002C     003B3                 66$-60$,-                          ;
                                                                   66$-60$,-                          ;
                                                                   62$-60$,-                          ;
                                                                   63$-60$,-                          ;
                                                                   66$-60$,-                          ;
                                                                   66$-60$,-                          ;
                                                                   64$-60$,-                          ;
                                                                   62$-60$                            ;
                      14  A7  0144    C8  9E 003B7 61$:    MOVAB   P.AFQ, ARGLIST+20                  ; 1245
                                      29  11 003BD         BRB     68$                                ;
                      14  A7  014E    C8  9E 003BF 62$:    MOVAB   P.AFR, ARGLIST+20                  ; 1248
                                      21  11 003C5         BRB     68$                                ;
                      14  A7  0157    C8  9E 003C7 63$:    MOVAB   P.AFS, ARGLIST+20                  ; 1251
                                      19  11 003CD         BRB     68$                                ;
                      14  A7  015F    C8  9E 003CF 64$:    MOVAB   P.AFT, ARGLIST+20                  ; 1254
                                      11  11 003D5 65$:    BRB     68$                                ;
                      10  A7  00C8    C7  D0 003D7 66$:    MOVL    UNKNOWN_DESC, ARGLIST+16           ; 1258
  14  A7          61      06          0A  EF 003DD         EXTZV   #10, #6, (R1), ARGLIST+20          ; 1259
                                      03  11 003E3         BRB     68$                                ; 1194
```

```
                              50            01 CE 003E5 67$:    MNEGL   #1, INDEX                              ; 1265
                                            50 D5 003E8 68$:    TSTL    INDEX                                  ; 1271
                                            07 15 003EA         BLEQ    69$                                    ;
                   10      A7     020C C740 7E 003EC            MOVAQ   CMDSTS_DESC_TBL[INDEX], ARGLIST+16     ; 1276
             FFFFFFFF      8F               50 D1 003F3 69$:    CMPL    INDEX, #-1                             ; 1280
                                            03 13 003FA         BEQL    70$                                    ;
                                          00A5 31 003FC         BRW     83$                                    ;
                   18      A7     016E      C8 9E 003FF 70$:    MOVAB   P.AFU, ARGLIST+24                      ; 1287
                                   50       01 CE 00405         MNEGL   #1, INDEX2                             ; 1288
      53           61              06       0A EF 00408         EXTZV   #10, #6, (R1), R3                      ; 1290
                   16              01       53 CF 0040D         CASEL   R3, #1, #22                            ;
    006C         0033            006C     002E    00411 71$:    .WORD   72$-71$,-                              ;
    006C         003C            006C     0037    00419                 80$-71$,-                              ;
    006C         006C            006C     006C    00421                 73$-71$,-                              ;
    006C         006C            006C     006C    00429                 80$-71$,-                              ;
    006C         004C            006C     0041    00431                 74$-71$,-                              ;
                 0061            006C     0056    00439                 80$-71$,-                              ;
                                                                        75$-71$,-                              ;
                                                                        80$-71$,-                              ;
                                                                        80$-71$,-                              ;
                                                                        80$-71$,-                              ;
                                                                        80$-71$,-                              ;
                                                                        80$-71$,-                              ;
                                                                        80$-71$,-                              ;
                                                                        80$-71$,-                              ;
                                                                        80$-71$,-                              ;
                                                                        80$-71$,-                              ;
                                                                        76$-71$,-                              ;
                                                                        80$-71$,-                              ;
                                                                        77$-71$,-                              ;
                                                                        80$-71$,-                              ;
                                                                        78$-71$,-                              ;
                                                                        80$-71$,-                              ;
                                                                        79$-71$                                ;
                              50            02 D0 0043F 72$:    MOVL    #2, INDEX2                             ; 1292
                                            3C 11 00442         BRB     81$                                    ;
                                            50 D4 00444 73$:    CLRL    INDEX2                                 ; 1293
                                            38 11 00446         BRB     81$                                    ;
                              50            01 D0 00448 74$:    MOVL    #1, INDEX2                             ; 1294
                                            33 11 0044B         BRB     81$                                    ;
                              50            03 D0 0044D 75$:    MOVL    #3, INDEX2                             ; 1295
                                            2E 11 00450         BRB     81$                                    ;
                              50            02 D0 00452 76$:    MOVL    #2, INDEX2                             ; 1298
                   18      A7     0177      C8 9E 00455         MOVAB   P.AFV, ARGLIST+24                      ; 1299
                                            23 11 0045B         BRB     81$                                    ; 1290
                                            50 D4 0045D 77$:    CLRL    INDEX2                                 ; 1304
                   18      A7     0195      C8 9E 0045F         MOVAB   P.AFW, ARGLIST+24                      ; 1305
                                            19 11 00465         BRB     81$                                    ; 1290
                              50            01 D0 00467 78$:    MOVL    #1, INDEX2                             ; 1310
                   18      A7     01B3      C8 9E 0046A         MOVAB   P.AFX, ARGLIST+24                      ; 1311
                                            0E 11 00470         BRB     81$                                    ; 1290
                              50            03 D0 00472 79$:    MOVL    #3, INDEX2                             ; 1316
                   18      A7     01D1      C8 9E 00475         MOVAB   P.AFY, ARGLIST+24                      ; 1317
                                            03 11 0047B         BRB     81$                                    ; 1290
                              50            01 CE 0047D 80$:    MNEGL   #1, INDEX2                             ; 1321
             FFFFFFFF      8F               50 D1 00480 81$:    CMPL    INDEX2, #-1                            ; 1327
                                            0F 13 00487         BEQL    82$                                    ;
```

```
                                10   A7      009C    C7  DO 00489          MOVL    FAO_STRINGS+48, ARGLIST+16                      ; 1333
                                14   A7      00AC  C740  DO 0048F          MOVL    MSG[INDEX2], ARGLIST+20                         ; 1334
                                                    0C  11 00496          BRB     83$                                             ; 1327
                                10   A7      00C8    C7  DO 00498 82$:     MOVL    UNKNOWN_DESC, ARGLIST+16                        ; 1342
    14   A7            61                          06  EF 0049E          EXTZV   #10, #6, (R1), ARGLIST+20                       ; 1343
                                04   AE      0200    8F  B0 004A4 83$:     MOVW    #512, REG_DESC                                  ; 1350
                                                  14   A7  9F 004AA          PUSHAB  ARGLIST+20                                      ; 1351
                                                  08   AE  9F 004AD          PUSHAB  REG_DESC
                                                  0C   AE  9F 004B0          PUSHAB  REG_DESC
                                                  10   A7  DD 004B3          PUSHL   ARGLIST+16
                        00000000G   00                    04  FB 004B6          CALLS   #4, SYS$FAOL
                              00C4  C7                    50  DO 004BD          MOVL    R0, STATUS
                                    0B                    50  E8 004C2          BLBS    R0, 84$
                                                  00C4  C7  DD 004C5          PUSHL   STATUS
                        00000000G   00                    01  FB 004C9          CALLS   #1, LIB$STOP
                                    50          0338    C7  DO 004D0 84$:     MOVL    CSR1, R0                                    ; 1357
        51          60              04                    00  EF 004D5          EXTZV   #0, #4, (R0), R1
                    0B              00                    51  CF 004DA          CASEL   R1, #0, #11
          0018          0064              0018                0018      004DE 85$:  .WORD   86$-85$,-
          0064          003B              003B                0064      004E6       86$-85$,-
          0064          003B              0018                0064      004EE       90$-85$,-
                                                                                    86$-85$,-
                                                                                    90$-85$,-
                                                                                    88$-85$,-
                                                                                    88$-85$,-
                                                                                    90$-85$,-
                                                                                    90$-85$,-
                                                                                    86$-85$,-
                                                                                    88$-85$,-
                                                                                    90$-85$
                                    50          0340    C7  DO 004F6 86$:     MOVL    CSR3, R0                                    ; 1367
                                    0B                    54  E8 004FB          BLBS    R4, 87$
    0C   A7            60                              02  0A EF 004FE          EXTZV   #10, #2, (R0), ARGLIST+12
                                    52                    08  DO 00504          MOVL    #8, STR_NUMBER                              ; 1368
                                                          3F  11 00507          BRB     91$                                         ; 1361
    0C   A7            60                              02  02 EF 00509 87$:     EXTZV   #2, #2, (R0), ARGLIST+12                    ; 1375
                                10   A7      04   AE  9E 0050F          MOVAB   REG_DESC, ARGLIST+16                                ; 1376
                                    52                    05  DO 00514          MOVL    #5, STR_NUMBER                              ; 1377
                                                          2F  11 00517          BRB     91$                                         ; 1357
                                    50          0340    C7  DO 00519 88$:     MOVL    CSR3, R0                                    ; 1390
                                    11                    54  E8 0051E          BLBS    R4, 89$
    0C   A7            60                              0A  00 EF 00521          EXTZV   #0, #10, (R0), ARGLIST+12
    10   A7            60                              02  0A EF 00527          EXTZV   #10, #2, (R0), ARGLIST+16                   ; 1391
                                    52                    06  DO 0052D          MOVL    #6, STR_NUMBER                              ; 1392
                                                          16  11 00530          BRB     91$                                         ; 1383
    0C   A7            60                              0A  00 EF 00532 89$:     EXTZV   #0, #10, (R0), ARGLIST+12                   ; 1399
                                10   A7      04   AE  9E 00538          MOVAB   REG_DESC, ARGLIST+16                                ; 1400
                                    52                    07  DO 0053D          MOVL    #7, STR_NUMBER                              ; 1401
                                                          06  11 00540          BRB     91$                                         ; 1357
                                    0D                    54  E9 00542 90$:     BLBC    R4, 92$                                     ; 1407
                                    52                    03  DO 00545          MOVL    #3, STR_NUMBER                              ; 1419
                                                  0C   A7  9F 00548 91$:     PUSHAB  ARGLIST+12                                  ; 1426
                                                6C A742    DD 0054B          PUSHL   FAO_STRINGS[STR_NUMBER]
                                    69                    02  FB 0054F          CALLS   #2, OUTPUT_LINES
                                                          04  00552 92$:     RET                                                 ; 1428

; Routine Size:  1363 bytes,    Routine Base:  $CODE + 0246
```

```
; 1016          1429  1
; 1017          1430  1 End
; 1018          1431  0 ELUDOM


                                                        .EXTRN  LIB$STOP

;                               PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $OWN$                         840  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  $PLIT                        4695  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  $CODE                        1945  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)


;                           Library Statistics
;
;                                    -------- Symbols --------      Pages       Processing
;       File                         Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1     18619        33         0      1000          00:02.0


;                                   COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:KMS3271/OBJ=OBJ$:KMS3271 MSRC$:KMS3271/UPDATE=(ENH$:KMS3271)

; 1019          1432  0
; Size:         1945 code + 5535 data bytes
; Run Time:       00:43.5
; Elapsed Time:   01:29.0
; Lines/CPU Min:   1974
; Lexemes/CPU-Min: 22699
; Memory Used:  345 pages
; Compilation Complete
```

GETCODE.LIS

INITPROC1.LIS

INITREAL.LIS

EXECIMAGE.LIS

ERFSUMM.LIS

INITBUS.LIS

INITPROC4.LIS

KMS3271.LIS

ERFTAPEVE.LIS

HEADER.LIS

FILES.LIS

ERLLOGSYS.LIS

INITPROC2.LIS

INIT_TAPE.LIS

ERLLOGMSG.LIS

INITDISK.LIS

INITPROC5.LIS

IMAGELOAD.LIS

INITPROC3.LIS

INTERVENE.LIS

ERFRTVEC.LIS

ERFSUMVEC.LIS

0150 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

LPA11K
LIS

MCHECKS
LIS

LCPRINTER
LIS

MBA
LIS

LOGMSCP
LIS

LINCHK
LIS

MBAINT
LIS

LABEL
LIS

LOGGER
LIS

MASSTAPE
LIS

MASSDISK
LIS