```
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
EEEEEEEEEEEEEFEE RRR          RRR   FFF
EEEEEEEEEEEEEEE  RRR          RRR   FFF
EEEEEEEEEEEEEEE  RRR          RRR   FFF
```

```
TTTTTTTTTT    AAAAAA    77777777      888888    DDDDDDDD    VV      VV  DDDDDDDD    EEEEEEEEEE  PPPPPPPP
TTTTTTTTTT    AAAAAA    77777777      888888    DDDDDDDD    VV      VV  DDDDDDDD    EEEEEEEEEE  PPPPPPPP
    TT       AA  AA          77      88  88    DD  DD      VV      VV  DD  DD      EE          PP  PP
    TT       AA  AA          77      88  88    DD  DD      VV      VV  DD  DD      EE          PP  PP
    TT       AA  AA          77      88  88    DD  DD      VV      VV  DD  DD      EE          PP  PP
    TT       AA  AA          77      88  88    DD  DD      VV      VV  DD  DD      EE          PP  PP
    TT       AA  AA        77        888888    DD  DD      VV      VV  DD  DD      EEEEEEEE    PPPPPPPP
    TT       AA  AA        77        888888    DD  DD      VV      VV  DD  DD      EEEEEEEE    PPPPPPPP
    TT       AAAAAAAAAA      77      88  88    DD  DD      VV      VV  DD  DD      EE          PP
    TT       AAAAAAAAAA      77      88  88    DD  DD      VV      VV  DD  DD      EE          PP
    TT       AA  AA        77        88  88    DD  DD        VV  VV    DD  DD      EE          PP
    TT       AA  AA        77        88  88    DD  DD        VV  VV    DD  DD      EE          PP
    TT       AA  AA      77          888888    DDDDDDDD        VV      DDDDDDDD    EEEEEEEEEE  PP
    TT       AA  AA      77          888888    DDDDDDDD        VV      DDDDDDDD    EEEEEEEEEE  PP

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


52

4F
52

20

4F
52

```
;     1          0001  0 MODULE TA78_DEVICE_DEPENDENT
;     2          0002  0 (%TITLE 'TA78 Device Dependent Module'
;     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 !*  COPYRIGHT (c) 1978, 1980, 1982 BY                                       *
;    10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
;    11          0011  1 !*  ALL RIGHTS RESERVED.                                                     *
;    12          0012  1 !*                                                                           *
;    13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    18          0018  1 !*  TRANSFERRED.                                                             *
;    19          0019  1 !*                                                                           *
;    20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    22          0022  1 !*  CORPORATION.                                                             *
;    23          0023  1 !*                                                                           *
;    24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
;    26          0026  1 !*                                                                           *
;    27          0027  1 !*                                                                           *
;    28          0028  1 !*****************************************************************************
;    29          0029  1
;    30          0030  1 !++
;    31          0031  1 ! FACILITY:  ERF, Error Log Report Generator
;    32          0032  1 !
;    33          0033  1 ! ABSTRACT:
;    34          0034  1 !
;    35          0035  1 !       This module contains the routines necessary to decode the TA78
;    36          0036  1 !       device dependent information that is appended to various
;    37          0037  1 !       'logmessage' entry formats.
;    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:  June-1984
;    44          0044  1 !
;    45          0045  1 !--
;    46          0046  1
;    47          0047  1 Require 'SRC$:ERFDEF.REQ' ;              ! ERF defintions
;    48          0333  1 Require 'SRC$:RECSELDEF.REQ' ;           ! Syecom definitions
;    49          0464  1
;    50          0465  1 External routine
;    51          0466  1     Translate_bits,
;    52          0467  1     Ots$powjj,
;    53          0468  1     Output_lines ;
;    54          0469  1
;    55          0470  1 External
;    56          0471  1     EMB:        $BBLOCK PSECT (EMB),
;    57          0472  1     SYECOM:     $BBLOCK PSECT (SYECOM) ;
```

```
;    58          0473  1
;    59          0474  1 Forward routine
;    60          0475  1     Cntrlr_flg_byte_decode : NOVALUE,
;    61          0476  1     Convert_bcd_number,
;    62          0477  1     Diagnostic_sts_byte_decode : NOVALUE,
;    63          0478  1     Error_number_byte_decode : NOVALUE,
;    64          0479  1     Label_and_hex_output : NOVALUE,
;    65          0480  1     Read_channel_tie_decode : NOVALUE,
;    66          0481  1     Summary_mode_byte_decode : NOVALUE,
;    67          0482  1     TA78_drive_sts_decode : NOVALUE,
;    68          0483  1     TA78_formatter_sts_decode : NOVALUE,
;    69          0484  1     TA78_unsucc_response_decode : NOVALUE ;
;    70          0485  1
;    71          0486  1 Own
;    72          0487  1     Arglist:            VECTOR [12,long],
;    73          0488  1     Bit_msk_0:          Initial (%X'FF'),
;    74          0489  1     Bit_msk_1:          Initial (%X'F0'),
;    75          0490  1     Bit_msk_2:          Initial (%X'F8'),
;    76          0491  1     Bit_msk_3:          Initial (%X'C0'),
;    77          0492  1     Bit_msk_4:          Initial (%X'70'),
;    78          0493  1     Bit_msk_5:          Initial (%X'1F'),
;    79          0494  1     Bit_msk_6:          Initial (%X'78'),
;    80          0495  1     Bit_msk_7:          Initial (%X'38'),
;    81          0496  1     Bit_msk_8:          Initial (%X'FE'),
;    82          0497  1     Bit_msk_9:          Initial (%X'E0'),
;    83          0498  1     Bit_msk_10:         Initial (%X'3C'),
;    84          0499  1     Bit_msk_11:         Initial (7),
;    85          0500  1     Bit_msk_12:         Initial (3),
;    86          0501  1     Fao_string,
;    87          0502  1     Fao_strings:        VECTOR [14,long]
;    88          0503  1                         Initial (long
;    89          0504  1                         (%ASCID '!/!AC'),
;    90          0505  1                         (%ASCID '!/!7< !>!AC!#< !>!XB!8< !>'),
;    91          0506  1                         (%ASCID '!39< !>!AS!ZB.'),
;    92          0507  1                         (%ASCID '!39< !>!AS'),
;    93          0508  1                         (%ASCID '!39< !>FAULT NUMBER INDICATES!/!39< !>POSSIBLE CAUSE GENERAL AREA =!/!39< !
;    94          0509  1                         (%ASCID '!39< !>PORT !AC, FORMATTER RECEPTION AND!/!39< !>TRANSMISSION ENABLED ON DA
;    95          0510  1                         (%ASCID '!39< !>PORT !AC, FORMATTER TRANSMISSION!/!39< !>ENABLED ON REAL TIME FORMAT
;    96          0511  1                         (%ASCID '!39< !>!AS!XB(X)'),
;    97          0512  1                         (%ASCID '!39< !>!ZB.!AS!/!39< !>!AS'),
;    98          0513  1                         (%ASCID '!/!7< !>!AC!ZB!#< !>!XB!8< !>'),
;    99          0514  1                         (%ASCID '!39< !>LAST CMD SENT TO M8953 VIA!/!39< !>"RCMD" = !AS'),
;   100          0515  1                         (%ASCID '!39< !>!ZB!AS'),
;   101          0516  1                         (%ASCID '!39< !>!AS!UW.'),
;   102          0517  1                         (%ASCID '!7< !>!AC!ZB!#< !>!XB!8< !>') ),
;   103          0518  1     Main_hdr:           byte,
;   104          0519  1     Out_arglist:        REF VECTOR [31,long],
;   105          0520  1     Sti_string:         Initial (%ASCID 'LAST STI LEVEL 2 CMD = ') ;
;   106          0521  1
;   107        P 0522  1 STORE_STRINGS ( byte31,
;   108          0523  1                 '0', '1', '2', '3', '4', '5', '6', '7' ) ;
;   109          0524  1
```

```
;  111          0525  1 GLOBAL Routine TA78_unsucc_response_decode  : NOVALUE =
;  112          0526  2 Begin
;  113          0527  2
;  114          0528  2 !++
;  115          0529  2 !
;  116          0530  2 ! Functional Description:
;  117          0531  2 !
;  118          0532  2 !       This routine decodes and outputs the unsuccessful response
;  119          0533  2 !       information that is appended to a logmessage 'STI error' log
;  120          0534  2 !       entry for a TA78.
;  121          0535  2 !
;  122          0536  2 ! Calling Sequence:
;  123          0537  2 !
;  124          0538  2 !       TA78_UNSUCC_RESPONSE_DECODE () ;
;  125          0539  2 !
;  126          0540  2 ! Input Parameters:
;  127          0541  2 !
;  128          0542  2 !       None.
;  129          0543  2 !
;  130          0544  2 ! Output Parameters:
;  131          0545  2 !
;  132          0546  2 !       None.
;  133          0547  2 !
;  134          0548  2 !--
;  135          0549  2
;  136          0550  2 Local
;  137          0551  2     J,
;  138          0552  2     K,
;  139          0553  2     Status ;
;  140          0554  2
;  141          0555  2
;  142        P 0556  2 STORE_STRINGS ( byte1,
;  143        P 0557  2                 'DIAGNOSTIC REQUESTED',
;  144        P 0558  2                 'PORT SWITCH ENABLED',
;  145        P 0559  2                 '"FORMATTER UNAVAILABLE" STATE',
;  146        P 0560  2                 'DRIVE ATTENTION FOR DRIVE ',
;  147          0561  2                 'FORMATTER ATTENTION' ) ;
;  148          0562  2
;  149        P 0563  2 STORE_STRINGS ( byte2,
;  150        P 0564  2                 'FORMATTER DIAGNOSTIC FAILED',
;  151        P 0565  2                 'LEVEL 2 PROTOCOL ERROR',
;  152        P 0566  2                 'LEVEL 1 TRANSMISSION ERROR',
;  153          0567  2                 'FORMATTER ERROR' ) ;
;  154          0568  2
;  155        P 0569  2 STORE_STRINGS ( byte5,
;  156        P 0570  2                 'ERROR LOGGING REQUEST',
;  157        P 0571  2                 'MAINTENANCE MODE REQUEST',
;  158        P 0572  2                 '"DRIVE AVAILABLE" STATE (TO FORMATTER)',
;  159        P 0573  2                 '"DRIVE ONLINE" STATE (TO FORMATTER)',
;  160        P 0574  2                 'WRITE LOCKED',
;  161        P 0575  2                 'BOT',
;  162        P 0576  2                 'EOT',
;  163          0577  2                 'TAPE MARK SEEN' ) ;
;  164          0578  2
;  165        P 0579  2 STORE_STRINGS ( byte6,
;  166        P 0580  2                 'DATA TRANSFER ERROR',
;  167        P 0581  2                 'EXCEPTION CONDITION DETECTED (TRANSFER)',
```

```
;  168        P 0582  2                  'POSITION LOST',
;  169        P 0583  2                  'LENGTHY OPERATION IN PROGRESS',
;  170          0584  2                  'DRIVE ERROR' ) ;
;  171          0585  2
;  172          0586  2
;  173          0587  2 Bind
;  174          0588  2     Unsucc_response = emb[82,0,8,0 ] : vector [,byte] ;
;  175          0589  2
;  176          0590  2 !
;  177          0591  2 ! Output the header and byte 1 (summary mode byte 1) information.
;  178          0592  2 !
;  179          0593  2 Arglist[0] = CSTRING ('UNSUCCESSFUL RESPONSE INFORMATION') ;
;  180          0594  2 Main_hdr = true ;
;  181          0595  2
;  182          0596  2 OUTPUT_LINES (  .fao_strings[0], arglist[0]  ) ;
;  183          0597  2
;  184          0598  2 LABEL_AND_HEX_OUTPUT (1) ;
;  185          0599  2
;  186          0600  2 !
;  187          0601  2 ! Translate bit to text.
;  188          0602  2 !
;  189          0603  3 If ( TRANSLATE_BITS (unsucc_response[0],bit_msk_0,
;  190          0604  3                      byte1_desc_tbl,out_arglist,
;  191          0605  3                      fao_string,%REF(0)) )
;  192          0606  2 Then
;  193          0607  2     ! Bit to tranlation done, output all of the information.
;  194          0608  2     !
;  195          0609  2     OUTPUT_LINES (  .fao_string, out_arglist[0]  ) ;
;  196          0610  2
;  197          0611  2 !
;  198          0612  2 ! Decode and output Byte 2 (summary error byte) information.
;  199          0613  2 !
;  200          0614  2 LABEL_AND_HEX_OUTPUT (2) ;
;  201          0615  2
;  202          0616  2 !
;  203          0617  2 ! Translate the bits to text.
;  204          0618  2 !
;  205          0619  3 If ( TRANSLATE_BITS (unsucc_response[1],bit_msk_1,byte2_desc_tbl,
;  206          0620  3                      out_arglist,fao_string,%REF(0)) )
;  207          0621  2 Then
;  208          0622  2     ! Bit to translation done, output everything.
;  209          0623  2     !
;  210          0624  2     OUTPUT_LINES (  .fao_string, out_arglist[0]  ) ;
;  211          0625  2
;  212          0626  2 !
;  213          0627  2 ! Decode and output Byte 3 (summary mode byte) information.
;  214          0628  2 !
;  215          0629  2 LABEL_AND_HEX_OUTPUT (3) ;
;  216          0630  2
;  217          0631  2 SUMMARY_MODE_BYTE_DECODE (.unsucc_response[2]) ;
;  218          0632  2
;  219          0633  2
;  220          0634  2 !
;  221          0635  2 ! Decode and output Byte 4 (controller byte) information.
;  222          0636  2 !
;  223          0637  2 LABEL_AND_HEX_OUTPUT (4) ;
;  224          0638  2
```

```
;  225          0639  2 CNTRLR_FLG_BYTE_DECODE (.unsucc_response[3]) ;
;  226          0640  2
;  227          0641  2
;  228          0642  2 !
;  229          0643  2 ! Decode and output the drive mode and drive error bytes.
;  230          0644  2 !
;  231          0645  2 ! Initialize the descriptor size and address.
;  232          0646  2 !
;  233          0647  2
;  234          0648  2 J = 5 ;
;  235          0649  2 K = 1 ;
;  236          0650  2 Incr I from 0 to 3 do
;  237          0651  3     Begin
;  238          0652  3     !
;  239          0653  3     ! Set up to decode and output the drive mode byte(s) information.
;  240          0654  3     !
;  241          0655  3     LABEL_AND_HEX_OUTPUT (.J) ;
;  242          0656  3
;  243          0657  3     !
;  244          0658  3     ! Translate the bits to text.
;  245          0659  3     !
;  246          0660  4     If ( TRANSLATE_BITS (unsucc_response[.J],bit_msk_0,
;  247          0661  4                          byte5_desc_tbl,out_arglist,
;  248          0662  4                          fao_string,%REF(0)) )
;  249          0663  3     Then
;  250          0664  3         ! Bit to text translation done, output all information.
;  251          0665  3         !
;  252          0666  3         OUTPUT_LINES (  .fao_string, out_arglist[0]  ) ;
;  253          0667  3
;  254          0668  3
;  255          0669  3     !
;  256          0670  3     ! Set up to decode and output the drive error byte(s) information.
;  257          0671  3     !
;  258          0672  3     LABEL_AND_HEX_OUTPUT (.J+1) ;
;  259          0673  3
;  260          0674  3     !
;  261          0675  3     ! Translate the bits to text.
;  262          0676  3     !
;  263          0677  4     If ( TRANSLATE_BITS (unsucc_response[.J],bit_msk_2,
;  264          0678  4                          byte6_desc_tbl,out_arglist,
;  265          0679  4                          fao_string,%REF(0)) )
;  266          0680  3     Then
;  267          0681  3         ! Bit to text translation done, output all information.
;  268          0682  3         !
;  269          0683  3         OUTPUT_LINES (  .fao_string, out_arglist[0]  ) ;
;  270          0684  3
;  271          0685  3     J = .J + 2 ;
;  272          0686  2     End ;
;  273          0687  2
;  274          0688  1 End ;    ! Routine


                                                .TITLE  TA78_DEVICE_DEPENDENT TA78 Device Dependent Mod
;                                                                                       ule
                                                .IDENT  \V04-000\
                                                .PSECT  $PLIT,NOWRT,NOEXE,  PIC,2
```

```
                                                00 00 00 43 41 21 2F 21  00000 P.AAB:  .ASCII  \!/!AC\<0><0><0>
                                                                010E0005 00008 P.AAA:  .LONG   17694725
                                                                00000000' 0000C        .ADDRESS P.AA8
20 3C 23 21 43 41 21 3E 21 20 3C 37 21 2F 21    00010 P.AAD:  .ASCII  \!/!7< !:AC!#< !>!XB!8< !>\<0><0>
      00 00 3E 21 20 3C 38 21 42 58 21 3E 21    0001F
                                                                010E001A 0002C P.AAC:  .LONG   17694746
                                                                00000000' 00030        .ADDRESS P.AAD
00 2E 42 5A 21 53 41 21 3E 21 20 3C 39 33 21    00034 P.AAF:  .ASCII  \!39< !>!AS!ZB.\<0><0>
                                                   00           00043
                                                                010E000E 00044 P.AAE:  .LONG   17694734
                                                                00000000' 00048        .ADDRESS P.AAF
            00 00 53 41 21 3E 21 20 3C 39 33 21    0004C P.AAH:  .ASCII  \!39< !>!AS\<0><0>
                                                                010E000A 00058 P.AAG:  .LONG   17694730
                                                                00000000' 0005C        .ADDRESS P.AAH
55 4E 20 54 4C 55 41 46 3E 21 20 3C 39 33 21    00060 P.AAJ:  .ASCII  \!39< !>FAULT NUMBER INDICATES!/!39< !>PO\
21 53 45 54 41 43 49 44 4E 49 20 52 45 42 4D    0006F
                  4F 50 3E 21 20 3C 39 33 21 2F 0007E
45 47 20 45 53 55 41 43 20 45 4C 42 49 53 53    00088         .ASCII  \SSIBLE CAUSE GENERAL AREA =!/!39< !>!AS-
21 2F 21 3D 20 41 45 52 41 20 4C 41 52 45 4E    00097               \<0>
                  00 53 41 21 3E 21 20 3C 39 33 000A6
                                                                010E004F 000B0 P.AAI:  .LONG   17694799
                                                                00000000' 000B4        .ADDRESS P.AAJ
43 41 21 20 54 52 4F 50 3E 21 20 3C 39 33 21    000B8 P.AAL:  .ASCII  \!39< !>PORT !AC, FORMATTER RECEPTION AND\
43 45 52 20 52 45 54 54 41 4D 52 4F 46 20 2C    000C7
                  44 4E 41 20 4E 4F 49 54 50 45 000D6
4D 53 4E 41 52 54 3E 21 20 3C 39 33 21 2F 21    000E0         .ASCII  \!/!39< !>TRANSMISSION ENABLED ON DATA'/!\
20 44 45 4C 42 41 4E 45 20 4E 4F 49 53 53 49    000EF
                  21 2F 21 41 54 41 44 20 4E 4F 000FE
3C 39 33 21 2F 21 53 41 21 3E 21 20 3C 39 33    00108         .ASCII  \39< !>.AS!/!39< !>!AS\<0><0><0>
                     00 00 00 53 41 21 3E 21 20 00117
                                                                010E0065 00120 P.AAK:  .LONG   17694821
                                                                00000000' 00124        .ADDRESS P.AAL
43 41 21 20 54 52 4F 50 3E 21 20 3C 39 33 21    00128 P.AAN:  .ASCII  \!39< !>PORT !AC, FORMATTER TRANSMISSION!\
41 52 54 20 52 45 54 54 41 4D 52 4F 46 20 2C    00137
                  21 4E 4F 49 53 53 49 4D 53 4E 00146
44 45 4C 42 41 4E 45 3E 21 20 3C 39 33 21 2F    00150         .ASCII  \/!39< !>ENABLED ON REAL TIME FORMATTER!/\
46 20 45 4D 49 54 20 4C 41 45 52 20 4E 4F 20    0015F
                  2F 21 52 45 54 54 41 4D 52 4F 0016E
49 4C 20 45 54 41 54 53 3E 21 20 3C 39 33 21    00178         .ASCII  \!39< !>STATE LINE\<0><0><0>
                                    00 00 00 45 4E 00187
                                                                010E0061 0018C P.AAM:  .LONG   17694817
                                                                00000000' 00190        .ADDRESS P.AAN
58 28 42 58 21 53 41 21 3E 21 20 3C 39 33 21    001[illegible] P.AAP:  .ASCII  \'39< !>!AS!XB(X)\
                                                   29           001A3
                                                                010E0010 001A4 P.AAO:  .LONG   17694736
                                                                00000000' 001A8        .ADDRESS P.AAP
21 53 41 21 2E 42 5A 21 3E 21 20 3C 39 33 21    001AC P.AAR:  .ASCII  \!39< !>!ZB.!AS'/!39< !>!AS\<0><0>
      00 00 53 41 21 3E 21 20 3C 39 33 21 2F    001BB
                                                                010E001A 001C8 P.AAQ:  .LONG   17694746
                                                                00000000' 001CC        .ADDRESS P.AAR
21 42 5A 21 43 41 21 3E 21 20 3C 37 21 2F 21    001D0 P.AAT:  .ASCII  \!/!7< !>!AC!ZB'#< !>!XB!8< !>\<0><0>
00 3E 21 20 3C 38 21 42 58 21 3E 21 20 3C 23    001DF
                                    00 00       001EE
                                    00          001EF         .ASCII  <0>
                                                                010E001D 001F0 P.AAS:  .LONG   17694749
                                                                00000000' 001F4        .ADDRESS P.AAT
```

```
44 4D 43 20 54 53 41 4C 3E 21 20 3C 39 33 21  001F8 P.AAV:  .ASCII  \!39< !>LAST CMD SENT TO M8953 VIA!/!39< \
20 33 35 39 38 4D 20 4F 54 20 54 4E 45 53 20  00207
                20 3C 39 33 21 2F 21 41 49 56  00216
00 53 41 21 20 3D 20 22 44 4D 43 52 22 3E 21  00220          .ASCII  \!>''RCMD'' = !AS\<0><0>
                                        00    0022F
                                  010E0036    00230 P.AAU:  .LONG   17694774
                                  00000000'   00234          .ADDRESS P.AAV
00 00 53 41 21 42 5A 21 3E 21 20 3C 39 33 21  00238 P.AAX:  .ASCII  \!39< !>!ZB!AS\<0><0><0>
                                        00    00247
                                  010E000D    00248 P.AAW:  .LONG   17694733
                                  00C00000'   0024C          .ADDRESS P.AAX
00 2E 57 55 21 53 41 21 3E 21 20 3C 39 33 21  00250 P.AAZ:  .ASCII  \!39< !>!AS!UW.\<0><0>
                                        00    0025F
                                  010E000E    00260 P.AAY:  .LONG   17694734
                                  00000000'   00264          .ADDRESS P.AAZ
3C 23 21 42 5A 21 43 41 21 3E 21 20 3C 37 21  00268 P.ABB:  .ASCII  \!7< !>!AC!ZB!#< !>!XB!8< !>\<0>
      00 3E 21 20 3C 38 21 42 58 21 3E 21 20  00277
                                  010E001B    00284 P.ABA:  .LONG   17694747
                                  00000000'   00288          .ADDRESS P.ABB
20 4C 45 56 45 4C 20 49 54 53 20 54 53 41 4C  0028C P.ABD:  .ASCII  \LAST STI LEVEL 2 CMD = \<0>
                  00 20 3D 20 44 4D 43 20 32  0029B
                                  010E0017    002A4 P.ABC:  .LONG   17694743
                                  00000000'   002A8          .ADDRESS P.ABD
                              00  00  00  30   002AC P.ABE:  .ASCII  \0\<0><0><0>
                              00  00  00  31   002B0 P.ABF:  .ASCII  \1\<0><0><0>
                              00  00  00  32   002B4 P.ABG:  .ASCII  \2\<0><0><0>
                              00  00  00  33   002B8 P.ABH:  .ASCII  \3\<0><0><0>
                              00  00  00  34   002BC P.ABI:  .ASCII  \4\<0><0><0>
                              00  00  00  35   002C0 P.ABJ:  .ASCII  \5\<0><0><0>
                              00  00  00  36   002C4 P.ABK:  .ASCII  \6\<0><0><0>
                              00  00  00  37   002C8 P.ABL:  .ASCII  \7\<0><0><0>
55 51 45 52 20 43 49 54 53 4F 4E 47 41 49 44  002CC P.ABM:  .ASCII  \DIAGNOSTIC REQ :STED\
                                    44 45 54 53 45  002DB
41 4E 45 20 48 43 54 49 57 53 20 54 52 4F 50  002E0 P.ABN:  .ASCII  \PORT SWITCH ENABLED\<0>
                                 00 44 45 4C 42  002EF
56 41 4E 55 20 52 45 54 54 41 4D 52 4F 46 22  002F4 P.ABO:  .ASCII  \''FORMATTER UNAVAILABLE'' STATE\<0><0>
00 45 54 41 54 53 20 22 45 4C 42 41 4C 49 41  00303
                                    00 00312
                                    00 00313          .ASCII  <0>
4E 4F 49 54 4E 45 54 54 41 20 45 56 49 52 44  00314 P.ABP:  .ASCII  \DRIVE ATTENTION FOR DRIVE \<0><0>
      00 00 20 45 56 49 52 44 20 52 4F 46 20  00323
4E 45 54 54 41 20 52 45 54 54 41 4D 52 4F 46  00330 P.ABQ:  .ASCII  \FORMATTER ATTENTION\<0>
                                 00 4E 4F 49 54  0033F
4E 47 41 49 44 20 52 45 54 54 41 4D 52 4F 46  00344 P.ABR:  .ASCII  \FORMATTER DIAGNOSTIC FAILED\<0>
      00 44 45 4C 49 41 46 20 43 49 54 53 4F  00353
4F 43 4F 54 4F 52 50 20 32 20 4C 45 56 45 4C  00360 P.ABS:  .ASCII  \LEVEL 2 PROTOCOL ERROR\<0><0>
                  00 00 52 4F 52 52 45 20 4C  0036F
49 4D 53 4E 41 52 54 20 31 20 4C 45 56 45 4C  00378 P.ABT:  .ASCII  \LEVEL 1 TRANSMISSION ERROR\<0><0>
      00 00 52 4F 52 52 45 20 4C 4F 49 53 53  00387
52 4F 52 52 45 20 52 45 54 54 41 4D 52 4F 46  00394 P.ABU:  .ASCII  \FORMATTER ERROR\<0>
                                    00 003A3
52 20 47 4E 49 47 47 4F 4C 20 52 4F 52 52 45  003A4 P.ABV:  .ASCII  \ERROR LOGGING REQUEST\<0><0><0>
                  00 00 00 54 53 45 55 51 45  003B3
44 4F 4D 20 45 43 4E 41 4E 45 54 4E 49 41 4D  003BC P.ABW:  .ASCII  \MAINTENANCE MODE REQUEST\
                  54 53 45 55 51 45 52 20 45  003CB
4C 42 41 4C 49 41 56 41 20 45 56 49 52 44 22  003D4 P.ABX:  .ASCII  \''DRIVE AVAILABLE'' STATE (TO FORMATTER)-
4F 46 20 4F 54 28 20 45 54 41 54 53 20 22 45  003E3          \<0>
```

```
                                  00 29 52 45 54 54 41 4D 52 003F2
                                                          00 003FB                .ASCII  <0>
20 22 45 4E 49 4C 4E 4F 20 45 56 49 52 44 22 003FC P.ABY:  .ASCII  \"DRIVE ONLINE" STATE (TO FORMATTER)\<0>
41 4D 52 4F 46 20 4F 54 28 20 45 54 41 54 53 00408
                                  00 29 52 45 54 54 00418
                  44 45 4B 43 4F 4C 20 45 54 49 52 57 00420 P.ABZ:  .ASCII  \WRITE LOCKED\
                                        00 54 4F 42 0042C P.ACA:  .ASCII  \BOT\<0>
                                        00 54 4F 45 00430 P.ACB:  .ASCII  \EOT\<0>
00 4E 45 45 53 20 4B 52 41 4D 20 45 50 41 54 00434 P.ACC:  .ASCII  \TAPE MARK SEEN\<0><0>
                                                          00 00443
45 20 52 45 46 53 4E 41 52 54 20 41 54 41 44 00444 P.ACD:  .ASCII  \DATA TRANSFER ERROR\<0>
                                        00 52 4F 52 52 00453
49 44 4E 4F 43 20 4E 4F 49 54 50 45 43 58 45 00458 P.ACE:  .ASCII  \EXCEPTION CONDITION DETECTED (TRANSFER)-
28 20 44 45 54 43 45 54 45 44 20 4E 4F 49 54 00467                \<0>
                  00 29 52 45 46 53 4E 41 52 54 00476
00 00 54 53 4F 4C 20 4E 4F 49 54 49 53 4F 50 00480 P.ACF:  .ASCII  \POSITION LOST\<0><0><0>
                                                          00 0048F
49 54 41 52 45 50 4F 20 59 48 54 47 4E 45 4C 00490 P.ACG:  .ASCII  \LENGTHY OPERATION IN PROGRESS\<0><0>
00 53 53 45 52 47 4F 52 50 20 4E 49 20 4E 4F 0049F
                                                          00 004AE
                                                          00 004AF                .ASCII  <0>
                  00 52 4F 52 52 45 20 45 56 49 52 44 004B0 P.ACH:  .ASCII  \DRIVE ERROR\<0>
52 20 4C 55 46 53 53 45 43 43 55 53 4E 55 21 004BC P.ACI:  .ASCII  \!UNSUCCESSFUL RESPONSE INFORMATION\
41 4D 52 4F 46 4E 49 20 45 53 4E 4F 50 53 45 004CB
                                        4E 4F 49 54 004DA

                                                                    .PSECT  $OWN$,NOEXE,  PIC,2

                                                00000 ARGLIST:.BLKB   48
000000FF                                        00030 BIT_MSK_0:
                                                                    .LONG   255
000000F0                                        00034 BIT_MSK_1:
                                                                    .LONG   240
000000F8                                        00038 BIT_MSK_2:
                                                                    .LONG   248
000000C0                                        0003C BIT_MSK_3:
                                                                    .LONG   192
00000070                                        00040 BIT_MSK_4:
                                                                    .LONG   112
0000001F                                        00044 BIT_MSK_5:
                                                                    .LONG   31
00000078                                        00048 BIT_MSK_6:
                                                                    .LONG   120
00000038                                        0004C BIT_MSK_7:
                                                                    .LONG   56
000000FE                                        00050 BIT_MSK_8:
                                                                    .LONG   254
000000E0                                        00054 BIT_MSK_9:
                                                                    .LONG   224
0000003C                                        00058 BIT_MSK_10:
                                                                    .LONG   60
00000007                                        0005C BIT_MSK_11:
                                                                    .LONG   7
00000003                                        00060 BIT_MSK_12:
                                                                    .LONG   3
                                                00064 FAO_STRING:
                                                                    .BLKB   4
```

```
                                        00000000' 00068 FAO_STRINGS:
                                                        .ADDRESS P.AAA
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 0006C       .ADDRESS P.AAC, P.AAE, P.AAG, P.AAI, P.AAK, -
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00084               P.AAM, P.AAO, P.AAQ, P.AAS, P.AAU, P.AAW, -
                                        00000000' 0009C                 P.AAY, P.ABA
                                                  000A0 MAIN_HDR:
                                                        .BLKB   1
                                                  000A1 .BLKB   3
                                                  000A4 OUT_ARGLIST:
                                                        .BLKB   4
                                        00000000' 000A8 STI_STRING:
                                                        .ADDRESS P.ABC
                                        00000001  000AC BYTE31_DESC_TBL:
                                                        .LONG   1
                                        00000000' 000B0 .ADDRESS P.ABE
                                        00000001  000B4 .LONG   1
                                        00000000' 000B8 .ADDRESS P.ABF
                                        00000001  000BC .LONG   1
                                        00000000' 000C0 .ADDRESS P.ABG
                                        00000001  000C4 .LONG   1
                                        00000000' 000C8 .ADDRESS P.ABH
                                        00000001  000CC .LONG   1
                                        00000000' 000D0 .ADDRESS P.ABI
                                        00000001  000D4 .LONG   1
                                        00000000' 000D8 .ADDRESS P.ABJ
                                        00000001  000DC .LONG   1
                                        00000000' 000E0 .ADDRESS P.ABK
                                        00000001  000E4 .LONG   1
                                        00000000' 000E8 .ADDRESS P.ABL
                                        00000014  000EC BYTE1_DESC_TBL:
                                                        .LONG   20
                                        00000000' 000F0 .ADDRESS P.ABM
                                        00000013  000F4 .LONG   19
                                        00000000' 000F8 .ADDRESS P.ABN
                                        0000001D  000FC .LONG   29
                                        00000000' 00100 .ADDRESS P.ABO
                                        0000001A  00104 .LONG   26
                                        00000000' 00108 .ADDRESS P.ABP
                                        00000013  0010C .LONG   19
                                        00000000' 00110 .ADDRESS P.ABQ
                                        0000001B  00114 BYTE2_DESC_TBL:
                                                        .LONG   27
                                        00000000' 00118 .ADDRESS P.ABR
                                        00000016  0011C .LONG   22
                                        00000000' 00120 .ADDRESS P.ABS
                                        0000001A  00124 .LONG   26
                                        00000000' 00128 .ADDRESS P.ABT
                                        0000000F  0012C .LONG   15
                                        00000000' 00130 .ADDRESS P.ABU
                                        00000015  00134 BYTE5_DESC_TBL:
                                                        .LONG   21
                                        00000000' 00138 .ADDRESS P.ABV
                                        00000018  0013C .LONG   24
                                        00000000' 00140 .ADDRESS P.ABW
                                        00000026  00144 .LONG   38
                                        00000000' 00148 .ADDRESS P.ABX
                                        00000023  0014C .LONG   35
```

```
                00000000' 00150          .ADDRESS P.ABY                                                  ;
                0000000C  00154          .LONG   12                                                      ;
                00000000' 00158          .ADDRESS P.ABZ                                                  ;
                00000003  0015C          .LONG   3                                                       ;
                00000000' 00160          .ADDRESS P.ACA                                                  ;
                00000003  00164          .LONG   3                                                       ;
                00000000' 00168          .ADDRESS P.ACB                                                  ;
                0000000E  0016C          .LONG   14                                                      ;
                00000000' 00170          .ADDRESS P.ACC                                                  ;
                00000013  00174 BYTE6_DESC_TBL:                                                          ;
                                         .LONG   19                                                      ;
                00000000' 00178          .ADDRESS P.ACD                                                  ;
                00000027  0017C          .LONG   39                                                      ;
                00000000' 00180          .ADDRESS P.ACE                                                  ;
                0000000D  00184          .LONG   13                                                      ;
                00000000' 00188          .ADDRESS P.ACF                                                  ;
                0000001D  0018C          .LONG   29                                                      ;
                00000000' 00190          .ADDRESS P.ACG                                                  ;
                0000000B  00194          .LONG   11                                                      ;
                00000000' 00198          .ADDRESS P.ACH                                                  ;

                                         .EXTRN  TRANSLATE_BITS, OTS$POWJJ
                                         .EXTRN  OUTPUT_LINES, EMB
                                         .EXTRN  SYECOM

                                         .PSECT  $CODE,NOWRT,  PIC,2

                      03FC 00000         .ENTRY  TA78_UNSUCC_RESPONSE_DECODE, Save R2,R3,R4,-; 0525
                                                 R5,R6,R7,R8,R9                                          ;
            59 00000000G   00   9E 00002         MOVAB   TRANSLATE_BITS, R9                              ;
            58 00000000G   00   9E 00009         MOVAB   UNSUCC_RESPONSE, R8                             ;
            57 00000000G   00   9E 00010         MOVAB   OUTPUT_LINES, R7                                ;
            56 00000000V   00   9E 00017         MOVAB   LABEL_AND_HEX_OUTPUT, R6                        ;
            55 00000000'   00   9E 0001E         MOVAB   FAO_STRING, R5                                  ;
            5E             04   C2 00025         SUBL2   #4, SP                                          ;
         9C A5 00000000'   00   9E 00028         MOVAB   P.ACI, ARGLIST                                  ; 0593
         3C A5             01   90 00030         MOVB    #1, MAIN_HDR                                    ; 0594
                 9C        A5   9F 00034         PUSHAB  ARGLIST                                         ; 0596
                 04        A5   DD 00037         PUSHL   FAO_STRINGS                                     ;
            67             02   FB 0003A         CALLS   #2, OUTPUT_LINES                                ;
                           01   DD 0003D         PUSHL   #1                                              ; 0598
            66             01   FB 0003F         CALLS   #1, LABEL_AND_HEX_OUTPUT                        ;
                           6E   D4 00042         CLRL    (SP)                                            ; 0605
                       4020     8F   BB 00044    PUSHR   #^M<R5,SP>                                      ; 0603
                         40     A5   9F 00048    PUSHAB  OUT_ARGLIST                                     ;
                       0088     C5   9F 0004B    PUSHAB  BYTE1_DESC_TBL                                  ;
                         CC     A5   9F 0004F    PUSHAB  BIT_MSK_0                                       ;
                                58   DD 00052    PUSHL   R8                                              ;
            69                   06   FB 00054    CALLS   #6, TRANSLATE_BITS                              ;
            08                   50   E8 00057    BLBC    R0, 1$                                          ;
                         40      A5   DD 0005A    PUSHL   OUT_ARGLIST                                     ; 0609
                                 65   DD 0005D    PUSHL   FAO_STRING                                      ;
            67                   02   FB 0005F    CALLS   #2, OUTPUT_LINES                                ;
                                 02   DD 00062 1$: PUSHL   #2                                            ; 0614
            66                   01   FB 00064    CALLS   #1, LABEL_AND_HEX_OUTPUT                        ;
                                 6E   D4 00067    CLRL    (SP)                                            ; 0620
                       4020      8F   BB 00069    PUSHR   #^M<R5,SP>                                      ; 0619
```

```
                                 40   A5  9F 0006D          PUSHAB  OUT_ARGLIST                                    ;
                               00B0   C5  9F 00070          PUSHAB  BYTE2_DESC_TBL                                 ;
                                 D0   A5  9F 00074          PUSHAB  BIT_MSK_1                                      ;
                                 01   A8  9F 00077          PUSHAB  UNSUCC_RESPONSE+1                              ;
                                      06  FB 0007A          CALLS   #6, TRANSLATE_BITS                             ;
                                      50  E9 0007D          BLBC    R0, 2$                                         ;
                                 40   A5  DD 00080          PUSHL   OUT_ARGLIST                                    ; 0624
                                      65  DD 00083          PUSHL   FAO_STRING                                     ;
                                      02  FB 00085          CALLS   #2, OUTPUT_LINES                               ;
                                      03  DD 00088 2$:      PUSHL   #3                                             ; 0629
                                      01  FB 0008A          CALLS   #1, LABEL_AND_HEX_OUTPUT                       ;
                                   02 7E  A8  9A 0008D      MOVZBL  UNSUCC_RESPONSE+2, -(SP)                       ; 0631
                         00000000v 00     01  FB 00091      CALLS   #1, SUMMARY_MODE_BYTE_DECODE                   ;
                                      04  DD 00098          PUSHL   #4                                             ; 0637
                                      01  FB 0009A          CALLS   #1, LABEL_AND_HEX_OUTPUT                       ;
                                   03 7E  A8  9A 0009D      MOVZBL  UNSUCC_RESPONSE+3, -(SP)                       ; 0639
                         00000000v 00     01  FB 000A1      CALLS   #1, CNTRLR_FLG_BYTE_DECODE                     ;
                                      52  D0 000A8          MOVL    #5, J                                          ; 0648
                                      50  D0 000AB          MOVL    #1, K                                          ; 0649
                                      54  D4 000AE          CLRL    I                                              ; 0650
                                      52  DD 000B0 3$:      PUSHL   J                                              ; 0655
                                      66  FB 000B2          CALLS   #1, LABEL_AND_HEX_OUTPUT                       ;
                                      6E  D4 000B5          CLRL    (SP)                                           ; 0662
                               4020   8F  BB 000B7          PUSHR   #^M<R5,SP>                                     ; 0650
                                 40   A5  9F 000BB          PUSHAB  OUT_ARGLIST                                    ;
                               00D0   C5  9F 000BE          PUSHAB  BYTE5_DESC_TBL                                 ;
                                 CC   A5  9F 000C2          PUSHAB  BIT_MSK_0                                      ;
                         53      52      C1 000C5          ADDL3   R8, J, R3                                      ;
                                      53  DD 000C9          PUSHL   R3                                             ;
                                      06  FB 000CB          CALLS   #6, TRANSLATE_BITS                             ;
                                      50  E9 000CE          BLBC    R0, 4$                                         ;
                                 40   A5  DD 000D1          PUSHL   OUT_ARGLIST                                    ; 0666
                                      65  DD 000D4          PUSHL   FAO_STRING                                     ;
                                      02  FB 000D6          CALLS   #2, OUTPUT_LINES                               ;
                                 01   A2  9F 000D9 4$:      PUSHAB  1(J)                                           ; 0672
                                      01  FB 000DC          CALLS   #1, LABEL_AND_HEX_OUTPUT                       ;
                                      6E  D4 000DF          CLRL    (SP)                                           ; 0679
                               4020   8F  BB 000E1          PUSHR   #^M<R5,SP>                                     ; 0677
                                 40   A5  9F 000E5          PUSHAB  OUT_ARGLIST                                    ;
                               0110   C5  9F 000E8          PUSHAB  BYTE6_DESC_TBL                                 ;
                                 D4   A5  9F 000EC          PUSHAB  BIT_MSK_2                                      ;
                                      53  DD 000EF          PUSHL   R3                                             ;
                                      06  FB 000F1          CALLS   #6, TRANSLATE_BITS                             ;
                                      50  E9 000F4          BLBC    R0, 5$                                         ;
                                 40   A5  DD 000F7          PUSHL   OUT_ARGLIST                                    ; 068?
                                      65  DD 000FA          PUSHL   FAO_STRING                                     ;
                                      02  FB 000FC          CALLS   #2, OUTPUT_LINES                               ;
                         52           52  C0 000FF 5$:      ADDL2   #2, J                                          ; 0685
                      AA 54           03  F3 00102          AOBLEQ  #3, I, 3$                                      ; 0650
                                          04 00106          RET                                                    ; 068R

; Routine Size:  263 bytes,    Routine Base:  $CODE + 0000

;  275          0689  1
```

```
;  277          0690  1 Routine SUMMARY_MODE_BYTE_DECODE (sum_mode_byte) : NOVALUE =
;  278          0691  2 Begin
;  279          0692  2 !++
;  280          0693  2 !
;  281          0694  2 ! Functional Description:
;  282          0695  2 !
;  283          0696  2 !       This routine decodes and outputs the summary mode byte. It is called
;  284          0697  2 !       from the TA78_unsucc_response_decode and the TA78_formatter_sts_decode
;  285          0698  2 !       routines.
;  286          0699  2 !
;  287          0700  2 ! Calling Sequence:
;  288          0701  2 !
;  289          0702  2 !       SUMMARY_MODE_BYTE_DECODE (sum_mode_byte) ;
;  290          0703  2 !
;  291          0704  2 ! Input Parameters:
;  292          0705  2 !
;  293          0706  2 !       Sum_mode_byte = contents of the summary mode byte location.
;  294          0707  2 !
;  295          0708  2 ! Output Parameters:
;  296          0709  2 !
;  297          0710  2 !       None.
;  298          0711  2 !
;  299          0712  2 !--
;  300        P 0713  2 STORE_STRINGS ( retry_bits,
;  301        P 0714  2                 'SAME',
;  302        P 0715  2                 'OPPOSITE',
;  303        P 0716  2                 'SUCCEEDED',
;  304        P 0717  2                 'FAILED',
;  305        P 0718  2                 'SAME',
;  306          0719  2                 'OPPOSITE') ;
;  307          0720  2
;  308          0721  2 Bind
;  309          0722  2     Retry_bits = .sum_mode_byte<0,2> ;
;  310          0723  2
;  311          0724  2 !
;  312          0725  2 ! Decode the retry status bits.
;  313          0726  2 !
;  314          0727  2 Arglist[0] = 0 ;
;  315          0728  2 Case retry_bits from 0 to 7 OF
;  316          0729  2     Set
;  317          0730  2     [0,1]: Fao_string = %ASCID '!39< !>NO ERROR' ;
;  318          0731  2
;  319          0732  2     [2,3]:
;  320          0733  3         Begin
;  321          0734  3         Fao_string = %ASCID '!39< !>READY FOR DATA TRANSFER IN !AS DIRECTION' ;
;  322          0735  3         Arglist[0] = retry_bits_desc_tbl[.retry_bits,0,0,0,0] ;
;  323          0736  2         End ;
;  324          0737  2
;  325          0738  2     [4,5]:
;  326          0739  3         Begin
;  327          0740  3         Fao_string = %ASCID '!39< !>RETRIED TRANSFER !AS' ;
;  328          0741  3         Arglist[0] = retry_bits_desc_tbl[.retry_bits,0,0,0,0] ;
;  329          0742  2         End ;
;  330          0743  2
;  331          0744  2     [6,7]:
;  332          0745  3         Begin
;  333          0746  3         Fao_string = %ASCID '!39< !>READY TO POSITION FOR RETRY IN !AS' ;
```

```
;  334          0747  3          Arglist[0] = retry_bits_desc_tbl[.retry_bits,0,0,0,0] ;
;  335          0748  2          End ;
;  336          0749  2
;  337          0750  2      [OUTRANGE]: fao_string = %ASCID '!39< !>UNKNOWN RETRY STATUS' ;
;  338          0751  2      Tes ;
;  339          0752  2
;  340          0753  2  !
;  341          0754  2  ! Output the information.
;  342          0755  2  !
;  343          0756  2  OUTPUT_LINES ( .fao_string, arglist[0] ) ;
;  344          0757  2
;  345          0758  2  !
;  346          0759  2  ! Decode bit 3 and output the necessary information.
;  347          0760  2  !
;  348          0761  2  If .sum_mode_byte<3>
;  349          0762  2  Then
;  350          0763  2      ! Bit 3 is set, get the text to output and output it.
;  351          0764  2      !
;  352          0765  3      Begin
;  353          0766  3      Arglist[1] = %ASCID 'ERROR LOGGING INFO AVAIL' ;
;  354          0767  3
;  355          0768  3      OUTPUT_LINES (  .fao_strings[3], arglist[1]  ) ;
;  356          0769  2      End ;
;  357          0770  2
;  358          0771  2  Return ;
;  359          0772  1  End ;   ! Routine


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

                                                        004DE           .BLKB   2
                                                45  4D  41  53 004E0 P.ACJ:  .ASC_I  \SAME\
                                            45  54  49  53  4F  50  50  4F 004E4 P.ACK:  .ASCII  \OPPOSITE\
                    00  00  00  44  45  44  45  45  43  43  55  53 004EC P.ACL:  .ASCII  \SUCCEEDED\<0><0><0>
                                        00  00  44  45  4C  49  41  46 004F8 P.ACM:  .ASCII  \FAILED\<0><0>
                                                45  4D  41  53 00500 P.ACN:  .ASCII  \SAME\
                                            45  54  49  53  4F  50  50  4F 00504 P.ACO:  .ASCII  \OPPOSITE\
52  4F  52  52  45  20  4F  4E  3E  21  20  3C  39  33  21 00050C P.ACQ:  .ASCII  \!39< !>NO ERROR\<0>
                                                        00  0051B
                                                010E000F    0051C P.ACP:  .LONG   17694735
                                                00000000'   00520         .ADDRESS P.ACQ
4F  46  20  59  44  41  45  52  3E  21  20  3C  39  33  21 00524 P.ACS:  .ASCII  \!39< !>READY FOR DATA TRANSFER IN !AS DI\
52  45  46  53  4E  41  52  54  20  41  54  41  44  20  52 00533
                        49  44  20  53  41  21  20  4E  49  20 00542
                                    00  4E  4F  49  54  43  45  52 0054C         .ASCII  \RECTION\<0>
                                                010E002F    00554 P.ACR:  .LONG   17694767
                                                00000000'   00558         .ADDRESS P.ACS
20  44  45  49  52  54  45  52  3E  21  20  3C  39  33  21 0055C P.ACU:  .ASCII  \!39< !>RETRIED TRANSFER !AS\<0>
        00  53  41  21  20  52  45  46  53  4E  41  52  54 00568
                                                010E001B    00578 P.ACT:  .LONG   17694747
                                                00000000'   0057C         .ADDRESS P.ACU
4F  54  20  59  44  41  45  52  3E  21  20  3C  39  33  21 00580 P.ACW:  .ASCII  \!39< !>READY TO POSITION FOR RETRY IN !A\
52  20  52  4F  46  20  4E  4F  49  54  49  53  4F  50  20 0058F
                        41  21  20  4E  49  20  53  52  54  45 0059E
                                        00  00  00  53 005A8         .ASCII  \S\<0><0><0>
                                                010E0029    005AC P.ACV:  .LONG   17694761
```

```
                                00000000'  005B0          .ADDRESS P.ACW
20 4E 57 4F 4E 4B 4E 55 3E 21 20 3C 39 33 21  005B4 P.ACY:  .ASCII  \!39< !>UNKNOWN RETRY STATUS\<0>
      00 53 55 54 41 54 53 20 59 52 54 45 52  005C3
                                010E001B  005D0 P.ACX:  .LONG   17694747
                                00000000'  005D4         .ADDRESS P.ACY
49 20 47 4E 49 47 47 4F 4C 20 52 4F 52 52 45  005D8 P.ADA:  .ASCII  \ERROR LOGGING INFO AVAIL\
                     4C 49 41 56 41 20 4F 46 4E  005E7
                                010E0018  005F0 P.ACZ:  .LONG   17694744
                                00000000'  005F4         .ADDRESS P.ADA

                                                         .PSECT  $OWN$,NOEXE,  PIC,2

                                00000004  0019C RETRY_BITS_DESC_TBL:
                                                         .LONG   4
                                00000000'  001A0         .ADDRESS P.ACJ
                                00000008  001A4          .LONG   8
                                00000000'  001A8         .ADDRESS P.ACK
                                00000009  001AC          .LONG   9
                                00000000'  001B0         .ADDRESS P.ACL
                                00000006  001B4          .LONG   6
                                00000000'  001B8         .ADDRESS P.ACM
                                00000004  001BC          .LONG   4
                                00000000'  001C0         .ADDRESS P.ACN
                                00000008  001C4          .LONG   8
                                00000000'  001C8         .ADDRESS P.ACO



                                                         .PSECT  $CODE,NOWRT,  PIC,2

                                001C 00000 SUMMARY_MODE_BYTE_DECODE:
                                                         .WORD   Save R2,R3,R4                                ; 0690
                          54 00000000G   00  9E 00002    MOVAB   OUTPUT_LINES, R4                             ;
                          53 00000000'   00  9E 00009    MOVAB   P.ACX, R3                                    ;
                          52 00000000'   00  9E 00010    MOVAB   FAO_STRING, R2                               ;
50      04   AC          02             00  EF 00017    EXTZV   #0, #2, SUM_MODE_BYTE, R0                    ; 0722
                    9C    A2             D4  0001D      CLRL    ARGLIST                                      ; 0727
        07          00                 50  CF 00020    CASEL   R0, #0, #7                                   ; 0728
001C        001C          0015             0015  00024 1$:  .WORD   2$-1$,-                                  ;
0028        0028          0022             0022  0002C              2$-1$,-                                  ;
                                                         3$-1$,-                                              ;
                                                         3$-1$,-                                              ;
                                                         4$-1$,-                                              ;
                                                         4$-1$,-                                              ;
                                                         5$-1$,-                                              ;
                                                         5$-1$                                                ;
                          62             63  9E 00034    MOVAB   P.ACX, FAO_STRING                            ; 0750
                                         21  11 00037    BRB     7$                                           ;
                          62     FF4C    C3  9E 00039 2$: MOVAB   P.ACP, FAO_STRING                           ; 0730
                                         1A  11 0003E    BRB     7$                                           ;
                          62       84    A3  9E 00040 3$: MOVAB   P.ACR, FAO_STRING                           ; 0734
                                         0A  11 00044    BRB     6$                                           ; 0735
                          62       A8    A3  9E 00046 4$: MOVAB   P.ACT, FAO_STRING                           ; 0740
                                         04  11 0004A    BRB     6$                                           ; 0741
                          62       DC    A3  9E 0004C 5$: MOVAB   P.ACV, FAO_STRING                           ; 0746
                          50             60  D0 00050 6$: MOVL    (R0), R0                                     ; 0747
                    9C    A2       0138 C240  7E 00053    MOVAQ   RETRY_BITS_DESC_TBL[R0], ARGLIST             ;
```

```
                                        9C  A2  9F 0005A 7$:     PUSHAB  ARGLIST                                 ; 0756
                                            62  DD 0005D         PUSHL   FAO_STRING                              ;
                                        64      02  FB 0005;     CALLS   #2,_OUTPUT_LINES                        ;
                                    0E      04  AC              03  E1 00062         BBC     #3, SUM_MODE_BYTE, 8$   ; 0761
                                            A0  A2          20  A3  9E 00067         MOVAB   P.ACZ, ARGLIST+4        ; 0766
                                                            A0  A2  9F 0006C         PUSHAB  ARGLIST+4               ; 0768
                                                            10  A2  DD 0006F         PUSHL   FAO_STRINGS+12          ;
                                                        64      02  FB 00072         CALLS   #2,_OUTPUT_LINES        ;
                                                                    04 00075 8$:     RET                             ; 0772

; Routine Size:  118 bytes,    Routine Base:  $CODE + 0107

;  360          0773  1
```

```
;  362     0774  1 Routine CNTRLR_FLG_BYTE_DECODE (cntrlr_flg_byte) : NOVALUE =
;  363     0775  2 Begin
;  364     0776  2
;  365     0777  2 !++
;  366     0778  2 !
;  367     0779  2 ! Functional Description:
;  368     0780  2 !
;  369     0781  2 !       This routine decodes and outputs the controller flag byte. It is
;  370     0782  2 !       called from the TA78_unsucc_response_decode and the
;  371     0783  2 !       TA78_formatter_sts_decode routines.
;  372     0784  2 !
;  373     0785  2 ! Calling Sequence:
;  374     0786  2 !
;  375     0787  2 !       CNTRLR_FLG_BYTE_DECODE (cntrlr_flg_byte) ;
;  376     0788  2 !
;  377     0789  2 ! Input Parameters:
;  378     0790  2 !
;  379     0791  2 !       Cntrlr_flg_byte = contents of the controller flag byte location.
;  380     0792  2 !
;  381     0793  2 ! Output Parameters:
;  382     0794  2 !
;  383     0795  2 !       None.
;  384     0796  2 !
;  385     0797  2 !--
;  386     0798  2
;  387     0799  2 Bind
;  388     0800  2     C1 = .cntrlr_flg_byte<0>,
;  389     0801  2     C2_thru_C8 = .cntrlr_flg_byte<1,7> ;
;  390     0802  2
;  391     0803  2
;  392     0804  2 Arglist[0] = %ASCID 'UNKNOWN CNTRLR FLAG BITS STATUS' ;
;  393     0805  2
;  394     0806  2 !
;  395     0807  2 ! Determine if the text associated with the controller bits.
;  396     0808  2 !
;  397     0809  2 If (C2_thru_C8) EQL 0
;  398     0810  2 Then
;  399     0811  3     Begin
;  400     0812  4     If NOT (C1)
;  401     0813  3     Then
;  402     0814  4         Begin
;  403     0815  4         Arglist[0] = %ASCID 'NORMAL OPERATION' ;
;  404     0816  4         End
;  405     0817  3     Else
;  406     0818  4         Begin
;  407     0819  5         If (C1)
;  408     0820  4         Then
;  409     0821  4             Arglist[0] = %ASCID 'DIAGNOSTIC MODE' ;
;  410     0822  3         End ;
;  411     0823  2     End ;
;  412     0824  2
;  413     0825  2 !
;  414     0826  2 ! Output the information.
;  415     0827  2 !
;  416     0828  2 OUTPUT_LINES (  .fao_strings[3], arglist[0]  ) ;
;  417     0829  2
;  418     0830  1 End ;   ! Routine
```

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

20 52 4C 52 54 4E 43 20 4E 57 4F 4E 4B 4E 55  005F8 P.ADC:   .ASCII  \UNKNOWN CNTRLR FLAG BITS STATUS\<0>                    ;
55 54 41 54 53 20 53 54 49 42 20 47 41 4C 46  00607                                                                       ;
                                              00  53  00616                                                               ;
                                                  010E001F  00618 P.ADB:   .LONG   17694751                               ;
                                                  00000000' 0061C          .ADDRESS P.ADC                                 ;
4F 49 54 41 52 45 50 4F 20 4C 41 4D 52 4F 4E  00620 P.ADE:   .ASCII  \NORMAL OPERATION\                                   ;
                                              4E  0062F                                                                   ;
                                                  010E0010  00630 P.ADD:   .LONG   17694736                               ;
                                                  00000000' 00634          .ADDRESS P.ADE                                 ;
45 44 4F 4D 20 43 49 54 53 4F 4E 47 41 49 44  00638 P.ADG:   .ASCII  \DIAGNOSTIC MODE\<0>                                ;
                                              00  00647                                                                   ;
                                                  010E000F  00648 P.ADF:   .LONG   17694735                               ;
                                                  00000000' 0064C          .ADDRESS P.ADG                                 ;


                                                                                .PSECT  $CODE,NOWRT,  PIC,2

                                                  000C 00000 CNTRLR_FLG_BYTE_DECODE:
                                                                .WORD   Save R2,R3                                      ; 0774
                                          53 00000000'  00  9E 00002            MOVAB   P.ADB, R3                       ;
                                          52 00000000'  00  9E 00009            MOVAB   ARGLIST, R2                     ;
                                          62            63  9E 00010            MOVAB   P.ADB, ARGLIST                  ; 0804
                                       FE 8F      04    AC  93 00013            BITB    CNTRLR_FLG_BYTE, #254           ; 0809
                                                        0E  12 00018            BNEQ    2$                              ;
                                          06        04    AC  E8 0001A            BLBS    CNTRLR_FLG_BYTE, 1$             ; 0812
                                          62        18    A3  9E 0001E            MOVAB   P.ADD, ARGLIST                  ; 0815
                                                        04  11 00022            BRB     2$                              ; 0812
                                          62        30    A3  9E 00024 1$:        MOVAB   P.ADF, ARGLIST                  ; 0821
                                                        52  DD 00028 2$:        PUSHL   R2                              ; 0828
                                                    74  A2  DD 0002A            PUSHL   FAO_STRINGS+12                  ;
                                  00000000G     00        02  FB 0002D            CALLS   #2, OUTPUT_LINES                ;
                                                            04 00034            RET                                     ; 0830

; Routine Size:  53 bytes,    Routine Base:  $CODE + 017D

;  419          0831  1
;  420          0832  1
```

```
;  422          0833  1 GLOBAL Routine TA78_FORMATTER_STS_DECODE : NOVALUE =
;  423          0834  2 Begin
;  424          0835  2 !++
;  425          0836  2 !
;  426          0837  2 ! Functional Description:
;  427          0838  2 !
;  428          0839  2 !       This routine decodes and outputs the extended formatter status response
;  429          0840  2 !       information that is appended to a logmessage 'STI formatter error'
;  430          0841  2 !       log entry for a TA78.
;  431          0842  2 !
;  432          0843  2 ! Calling Sequence:
;  433          0844  2 !
;  434          0845  2 !       TA78_FORMATTER_STS_DECODE () ;
;  435          0846  2 !
;  436          0847  2 ! Input Parameters:
;  437          0848  2 !
;  438          0849  2 !       None.
;  439          0850  2 !
;  440          0851  2 ! Output Parameters:
;  441          0852  2 !
;  442          0853  2 !       None.
;  443          0854  2 !
;  444          0855  2 !--
;  445          0856  2
;  446          0857  2
;  447        P 0858  2 STORE_STRINGS ( byte4,
;  448        P 0859  2                 'FORMATTER ONLINE TO CONTROLLER',
;  449        P 0860  2                 'FORMATTER AVAILABLE TO CONTROLLER',
;  450        P 0861  2                 'FORMATTER IN TOPOLOGY MODE',
;  451        P 0862  2                 'FORMATTER OFFLINE TO CONTROLLER',
;  452          0863  2                 'INTERNAL HARDWARE PROBLEM' ) ;
;  453          0864  2
;  454        P 0865  2 STORE_STRINGS ( byte5,
;  455        P 0866  2                 'FORMATTER DIAGNOSTIC REQUEST',
;  456        P 0867  2                 'PORT A/B ENABLED',
;  457        P 0868  2                 'FORMATTER IN TOPOLOG' MODE',
;  458        P 0869  2                 'TRANSPORT 0 ATTENTION',
;  459        P 0870  2                 'TRANSPORT 1 ATTENTION',
;  460        P 0871  2                 'TRANSPORT 2 ATTENTION',
;  461        P 0872  2                 'TRANSPORT 3 ATTENTION',
;  462          0873  2                 'FORMATTER ATTENTION' ) ;
;  463          0874  2
;  464        P 0875  2 STORE_STRINGS ( byte6,
;  465        P 0876  2                 'DIAGNOSTIC FAILED',
;  466        P 0877  2                 'LEVEL 2 PROTOCOL ERROR',
;  467        P 0878  2                 'TRANSMISSION ERROR',
;  468          0879  2                 'FORMATTER ERROR (OPERATIONAL CODE)' ) ;
;  469          0880  2
;  470        P 0881  2 STORE_STRINGS ( byte10,
;  471        P 0882  2                 'DATA READY',
;  472        P 0883  2                 'ACKNOWLEDGE',
;  473        P 0884  2                 'ATTENTION',
;  474          0885  2                 'FORMATTER RECEIVER READY' ) ;
;  475          0886  2
;  476        P 0887  2 STORE_STRINGS ( byte11,
;  477        P 0888  2                 'PORT SELECT B',
;  478        P 0889  2                 'PORT SELECT A',
```

```
;  479          0890  2                    'FAULT PRESSED' ) ;
;  480          0891  2
;  481        P 0892  2 STORE_STRINGS ( byte13,
;  482        P 0893  2                 'MOVED TAPE IN OPPOSITE DIRECTION',
;  483        P 0894  2                 'MOVED TAPE IN REVERSE DIRECTION',
;  484          0895  2                 'WAS A WRITE' ) ;
;  485          0896  2
;  486          0897  2 Bind
;  487          0898  2     Frmtr_sts = emb[82,0,8,0] : VECTOR [,byte],
;  488          0899  2     Errnum = frmtr_sts[0] : word ;
;  489          0900  2
;  490          0901  2
;  491          0902  2 !
;  492          0903  2 ! Output the header and decode and output bytes 1 and 2, the error number.
;  493          0904  2 !
;  494          0905  2 Arglist[0] = CSTRING ('TA78 EXTENDED FORMATTER STATUS') ;
;  495          0906  2 Main_hdr = true ;
;  496          0907  2 OUTPUT_LINES (  .fao_strings[0], arglist[0]  ) ;
;  497          0908  2
;  498          0909  2 LABEL_AND_HEX_OUTPUT (1,2) ;
;  499          0910  2
;  500          0911  2 ERROR_NUMBER_BYTE_DECODE (.errnum) ;
;  501          0912  2
;  502          0913  2 !
;  503          0914  2 ! Decode and output byte 3, last received level 2 opcode.
;  504          0915  2 !
;  505          0916  2 LABEL_AND_HEX_OUTPUT (3) ;
;  506          0917  2 Arglist[0] = .sti_string ;
;  507          0918  2 Arglist[1] = .frmtr_sts[2] ;
;  508          0919  2
;  509          0920  2 OUTPUT_LINES (  .fao_strings[7], arglist[0]  ) ;
;  510          0921  2
;  511          0922  2 !
;  512          0923  2 ! Decode and output byte 4, connection state byte.
;  513          0924  2 !
;  514          0925  2 LABEL_AND_HEX_OUTPUT (4) ;
;  515          0926  2
;  516          0927  3 If ( TRANSLATE_BITS (frmtr_sts[3],bit_msk_5,byte4_desc_tbl,
;  517          0928  3                        out_arglist,fao_string,%REF(0)) )
;  518          0929  2 Then
;  519          0930  2     OUTPUT_LINES ( .fao_string, out_arglist[0] ) ;
;  520          0931  2
;  521          0932  2 !
;  522          0933  2 ! Decode and output byte 5, summary mode byte 1.
;  523          0934  2 !
;  524          0935  2 LABEL_AND_HEX_OUTPUT (5) ;
;  525          0936  2
;  526          0937  3 If ( TRANSLATE_BITS (frmtr_sts[4],bit_msk_0,byte5_desc_tbl,
;  527          0938  3                        out_arglist,fao_string,%REF(0)) )
;  528          0939  2 Then
;  529          0940  2     OUTPUT_LINES ( .fao_string, out_arglist[0] ) ;
;  530          0941  2
;  531          0942  2
;  532          0943  2 !
;  533          0944  2 ! Decode and output byte 6, summary error byte.
;  534          0945  2 !
;  535          0946  2 LABEL_AND_HEX_OUTPUT (6) ;
```

```
;  536     0947  2
;  537     0948  3 If ( TRANSLATE_BITS (frmtr_sts[5],bit_msk_1,byte6_desc_tbl,
;  538     0949  3                         out_arglist,fao_string,%REF(0)) )
;  539     0950  2 Then
;  540     0951  2     OUTPUT_LINES ( .fao_string, out_arglist[0] ) ;
;  541     0952  2
;  542     0953  2
;  543     0954  2 !
;  544     0955  2 ! Decode and output byte 7, summary mode byte 2.
;  545     0956  2 !
;  546     0957  2 LABEL_AND_HEX_OUTPUT (7) ;
;  547     0958  2
;  548     0959  2 SUMMARY_MODE_BYTE_DECODE (.frmtr_sts[6]) ;
;  549     0960  2
;  550     0961  2
;  551     0962  2 !
;  552     0963  2 ! Decode and output byte 8, controller flag byte.
;  553     0964  2 !
;  554     0965  2 LABEL_AND_HEX_OUTPUT (8) ;
;  555     0966  2
;  556     0967  2 CNTRLR_FLG_BYTE_DECODE (.frmtr_sts[7]) ;
;  557     0968  2
;  558     0969  2 !
;  559     0970  2 ! Decode and output byte 9, port control byte.
;  560     0971  2 !
;  561     0972  2 LABEL_AND_HEX_OUTPUT (9) ;
;  562     0973  2
;  563     0974  2 If .(frmtr_sts[8])<0> OR
;  564     0975  2     .(frmtr_sts[8])<1>
;  565     0976  2 Then
;  566     0977  3     Begin
;  567     0978  3     Arglist[4] = %ASCID 'LINES FORMATTER RECEPTION ENABLED' ;
;  568     0979  3     Arglist[5] = %ASCID 'ON REALTIME CONTROLLER STATE LINE' ;
;  569     0980  3     fao_string = .fao_strings[5] ;
;  570     0981  2     End ;
;  571     0982  2
;  572     0983  2 If .(frmtr_sts[8])<2> OR
;  573     0984  2     .(frmtr_sts[8])<3>
;  574     0985  2 Then
;  575     0986  2     fao_string = .fao_strings[6] ;
;  576     0987  2
;  577     0988  2 If .(frmtr_sts[8])<0> OR
;  578     0989  2     .(frmtr_sts[8])<2>
;  579     0990  2 Then
;  580     0991  2     Arglist[3] = CSTRING ('B') ;
;  581     0992  2
;  582     0993  2 If .(frmtr_sts[8])<1> OR
;  583     0994  2     .(frmtr_sts[8])<3>
;  584     0995  2 Then
;  585     0996  2     Arglist[3] = CSTRING ('A') ;
;  586     0997  2
;  587     0998  2
;  588     0999  2 OUTPUT_LINES (  .fao_string, arglist[3] ) ;
;  589     1000  2
;  590     1001  2
;  591     1002  2
;  592     1003  2 ! Decode and output byte 10, formatter state byte.
```

```
;  593          1004  2 !
;  594          1005  2 LABEL_AND_HEX_OUTPUT (10) ;
;  595          1006  2
;  596          1007  2 If .(frmtr_sts[9])<0>
;  597          1008  2 Then
;  598          1009  2     Arglist[0] = %ASCID 'FORCE PARITY ERROR'
;  599          1010  2 Else
;  600          1011  2     Arglist[0] = %ASCID 'NORMAL OPERATION' ;
;  601          1012  2
;  602          1013  2 If NOT .(frmtr_sts[9])<1>
;  603          1014  2 Then
;  604          1015  2     Arglist[1] = %ASCID 'FORMATTER ONLINE'
;  605          1016  2 Else
;  606          1017  2     Arglist[1] = %ASCID 'FORMATTER AVAILABLE OR OFFLINE' ;
;  607          1018  2
;  608          1019  2
;  609          1020  2 If .(frmtr_sts[3])<0>           ! Formatter online
;  610          1021  2 Then
;  611          1022  3     Begin
;  612          1023  4     If ( TRANSLATE_BI ( (frmtr_sts[9],bit_msk_6,byte10_desc_tbl,
;  613          1024  4                          out_arglist,fao_string,%REF(0)) )
;  614          1025  3     Then
;  615          1026  3         OUTPUT_LINES (  .fao_strings[3], arglist[0],
;  616          1027  3                         .fao_strings[3], arglist[1],
;  617          1028  3                         .fao_string, out_arglist[0]  ) ;
;  618          1029  3     End
;  619          1030  2 Else
;  620          1031  3     Begin
;  621          1032  3     If .(frmtr_sts[3])<1>              ! Formatter available
;  622          1033  3     Then
;  623          1034  4         Begin
;  624          1035  4         Fao_string = %ASCID 'TRANSPORT CONNECTION STATE CHANGE!/!39< !>CNT = !ZB' ;
;  625          1036  4         Arglist[3] = .(frmtr_sts[9])<2,5> ;
;  626          1037  4
;  627          1038  4         OUTPUT_LINES (  .fao_strings[3], arglist[0],
;  628          1039  4                         .fao_strings[3], arglist[1],
;  629          1040  4                         .fao_string, arglist[3]  ) ;
;  630          1041  4
;  631          1042  4         If .(frmtr_sts[9])<6>
;  632          1043  4         Then
;  633          1044  5             Begin
;  634          1045  5             Arglist[3] = byte10_desc_tbl[6,0,0,0,0] ;
;  635          1046  5             OUTPUT_LINES ( .fao_strings[3], arglist[3] ) ;
;  636          1047  4             End ;
;  637          1048  3         End ;
;  638          1049  2     End ;
;  639          1050  2
;  640          1051  2
;  641          1052  2 If .(frmtr_sts[9])<7>
;  642          1053  2 Then
;  643          1054  2     Fao_string = %ASCID '!39< !>ALLOWS TRANSMISSION OF STATE!/!39< !>BITS ON ENABLED PORT(S) REALTIME!/!39<
;  644          1055  2 Else
;  645          1056  2     Fao_string = %ASCID '!39< !>FORCE TRANSMISSION OF ZEROS ON!/!39< !>ENABLED PORT(S) REALTIME!/'39< !>FORM
;  646          1057  2
;  647          1058  2 OUTPUT_LINES (  .fao_string, %REF(0) ) ;
;  648          1059  2
;  649          1060  2 !
```

```
;  650          1061  2 ! Decode and output byte 11, port switch byte.
;  651          1062  2 !
;  652          1063  2 LABEL_AND_HEX_OUTPUT (11) ;
;  653          1064  2
;  654          1065  3 If ( TRANSLATE_BITS (frmtr_sts[10],bit_msk_4,byte11_desc_tbl,
;  655          1066  3                          out_arglist,fao_string,%REF(0)) )
;  656          1067  2 Then
;  657          1068  2     OUTPUT_LINES ( .fao_string, out_arglist[0] ) ;
;  658          1069  2
;  659          1070  2
;  660          1071  2 !
;  661          1072  2 ! Decode and output byte 12, last status code.
;  662          1073  2 !
;  663          1074  2 LABEL_AND_HEX_OUTPUT (12) ;
;  664          1075  2
;  665          1076  2 Arglist[0] = %ASCID 'LAST STATUS CODE = ' ;
;  666          1077  2 Arglist[1] = .frmtr_sts[11] ;
;  667          1078  2
;  668          1079  2 OUTPUT_LINES (  .fao_strings[7], arglist[0]  ) ;
;  669          1080  2
;  670          1081  2 !
;  671          1082  2 ! Decode and output byte 13, error retry flag byte.
;  672          1083  2 !
;  673          1084  2 LABEL_AND_HEX_OUTPUT (13) ;
;  674          1085  2
;  675          1086  2 Arglist[0] = %ASCID 'UNIT = ' ;
;  676          1087  2 Arglist[1] = .(frmtr_sts[12])<1,2> ;
;  677          1088  2
;  678          1089  2 OUTPUT_LINES (  .fao_strings[2], arglist[0]  ) ;
;  679          1090  2
;  680          1091  2 If .(frmtr_sts[12])<3,5> NEQ 0
;  681          1092  2 Then
;  682          1093  3     Begin
;  683          1094  3     Arglist[2] = %ASCID 'INITIAL COMMAND -' ;
;  684          1095  3     OUTPUT_LINES (  .fao_strings[3], arglist[2]  ) ;
;  685          1096  3
;  686          1097  4     If ( TRANSLATE_BITS (frmtr_sts[12],bit_msk_7,byte13_desc_tbl,
;  687          1098  4                          out_arglist,fao_string,%ref(0)) )
;  688          1099  3     Then
;  689          1100  3         OUTPUT_LINES ( .fao_string, out_arglist[0] ) ;
;  690          1101  2     End ;
;  691          1102  2
;  692          1103  2 !
;  693          1104  2 ! Decode and output byte 14, retry counter.
;  694          1105  2 !
;  695          1106  2 LABEL_AND_HEX_OUTPUT (14) ;
;  696          1107  2
;  697          1108  2 Arglist[0] = .frmtr_sts[13] ;
;  698          1109  2 Arglist[1] = %ASCID ' RETRY REQUESTS DURING ERROR' ;
;  699          1110  2 Arglist[2] = %ASCID 'RECOVERY SEQUENCE' ;
;  700          1111  2
;  701          1112  2 OUTPUT_LINES (  .fao_strings[8], arglist[0]  ) ;
;  702          1113  2
;  703          1114  2 !
;  704          1115  2 ! Decode and output byte 15, STI bus init counter.
;  705          1116  2 !
;  706          1117  2 LABEL_AND_HEX_OUTPUT (15) ;
```

```
;  707          1118  2
;  708          1119  2 Arglist[0] = .frmtr_sts[14] ;
;  709          1120  2 Arglist[1] = %ASCID ' INITS SINCE TA78 MASTER RESET' ;
;  710          1121  2 Arglist[2] = %ASCID 'OR PWR UP' ;
;  711          1122  2
;  712          1123  2 OUTPUT_LINES (  .fao_strings[8], arglist[0]  ) ;
;  713          1124  2
;  714          1125  2 !
;  715          1126  2 ! Decode and output byte 16, parameter address.
;  716          1127  2 !
;  717          1128  2 LABEL_AND_HEX_OUTPUT (16) ;
;  718          1129  2
;  719          1130  2 If .errnum EQL %X'3C5F'                  ! Underflow error
;  720          1131  2 Then
;  721          1132  3     Begin
;  722          1133  3     Arglist[0] = %ASCID 'CONTENTS OF ADDRESS IN (SP)' ;
;  723          1134  3
;  724          1135  3     OUTPUT_LINES (  .fao_strings[3], arglist[0]  ) ;
;  725          1136  2     End ;
;  726          1137  2
;  727          1138  2
;  728          1139  2 !
;  729          1140  2 ! Decode and output byte 17, parameter address.
;  730          1141  2 !
;  731          1142  2 LABEL_AND_HEX_OUTPUT (17) ;
;  732          1143  2
;  733          1144  2 If .errnum EQL %X'3C5F'                  ! Underflow error
;  734          1145  2 Then
;  735          1146  3     Begin
;  736          1147  3     Arglist[0] = %ASCID 'CONTENTS OF ADDRESS IN (SP+1)' ;
;  737          1148  3
;  738          1149  3     OUTPUT_LINES (  .fao_strings[3], arglist[0]  ) ;
;  739          1150  2     End ;
;  740          1151  2
;  741          1152  2
;  742          1153  2 !
;  743          1154  2 ! Decode and output byte 18, parameter address.
;  744          1155  2 !
;  745          1156  2 LABEL_AND_HEX_OUTPUT (18) ;
;  746          1157  2
;  747          1158  2 If .errnum EQL %X'3C5F'                  ! Underflow error
;  748          1159  2 Then
;  749          1160  3     Begin
;  750          1161  3     Arglist[0] = %ASCID 'CONTENTS OF ADDRESS IN (SP+2)' ;
;  751          1162  3
;  752          1163  3     OUTPUT_LINES (  .fao_strings[3], arglist[0]  ) ;
;  753          1164  2     End ;
;  754          1165  2
;  755          1166  2
;  756          1167  2 !
;  757          1168  2 ! Decode and output byte 19, parameter address.
;  758          1169  2 !
;  759          1170  2 LABEL_AND_HEX_OUTPUT (19) ;
;  760          1171  2
;  761          1172  2 If .errnum EQL %X'3C5F'                  ! Underflow error
;  762          1173  2 Then
;  763          1174  3     Begin
```

```
;  764          1175  3     Arglist[0] = %ASCID 'CONTENTS OF ADDRESS IN (SP+3)' ;
;  765          1176  3
;  766          1177  3     OUTPUT_LINES (  .fao_strings[3], arglist[0]  ) ;
;  767          1178  2     End ;
;  768          1179  2
;  769          1180  2
;  770          1181  2 !
;  771          1182  2 ! Byte 20, spare.
;  772          1183  2 !
;  773          1184  2 LABEL_AND_HEX_OUTPUT (20) ;
;  774          1185  2
;  775          1186  1 End ;   ! Routine


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

4E 49 4C 4E 4F 20 52 45 54 54 41 4D 52 4F 46  00650 P.ADH:   .ASCII  \FORMATTER ONLINE TO CONTROLLER\<0><0>
52 45 4C 4C 4F 52 54 4E 4F 43 20 4F 54 20 45  0065F
                                              00 00  0066E
4C 49 41 56 41 20 52 45 54 54 41 4D 52 4F 46  00670 P.ADI:   .ASCII  \FORMATTER AVAILABLE TO CONTROLLER\<0><0>
4C 4F 52 54 4E 4F 43 20 4F 54 20 45 4C 42 41  0067F
                                  00 00 52 45 4C  0068E
                                        00  00693           .ASCII  <0>
4F 54 20 4E 49 20 52 45 54 54 41 4D 52 4F 46  00694 P.ADJ:   .ASCII  \FORMATTER IN TOPOLOGY MODE\<0><0>
   00 00 45 44 4F 4D 20 59 47 4F 4C 4F 50     006A3
49 4C 46 46 4F 20 52 45 54 54 41 4D 52 4F 46  006B0 P.ADK:   .ASCII  \FORMATTER OFFLINE TO CONTROLLER\<0>
45 4C 4C 4F 52 54 4E 4F 43 20 4F 54 20 45 4E  006BF
                                        00 52  006CE
41 57 44 52 41 48 20 4C 41 4E 52 45 54 4E 49  006D0 P.ADL:   .ASCII  \INTERNAL HARDWARE PROBLEM\<0><0><0>
   00 00 00 4D 45 4C 42 4F 52 50 20 45 52     006DF
4E 47 41 49 44 20 52 45 54 54 41 4D 52 4F 46  006EC P.ADM:   .ASCII  \FORMATTER DIAGNOSTIC REQUEST\
   54 53 45 55 51 45 52 20 43 49 54 53 4F     006FB
45 4C 42 41 4E 45 20 42 2F 41 20 54 52 4F 50  00708 P.ADN:   .ASCII  \PORT A/B ENABLED\
                                              44  00717
4F 54 20 4E 49 20 52 45 54 54 41 4D 52 4F 46  00718 P.ADO:   .ASCII  \FORMATTER IN TOPOLOGY MODE\<0><0>
   00 00 45 44 4F 4D 20 59 47 4F 4C 4F 50     00727
54 54 41 20 30 20 54 52 4F 50 53 4E 41 52 54  00734 P.ADP:   .ASCII  \TRANSPORT 0 ATTENTION\<0><0><0>
                  00 00 00 4E 4F 49 54 4E 45  00743
54 54 41 20 31 20 54 52 4F 50 53 4E 41 52 54  0074C P.ADQ:   .ASCII  \TRANSPORT 1 ATTENTION\<0><0><0>
                  00 00 00 4E 4F 49 54 4E 45  0075B
54 54 41 20 32 20 54 52 4F 50 53 4E 41 52 54  00764 P.ADR:   .ASCII  \TRANSPORT 2 ATTENTION\<0><0><0>
                  00 00 00 4E 4F 49 54 4E 45  00773
54 54 41 20 33 20 54 52 4F 50 53 4E 41 52 54  0077C P.ADS:   .ASCII  \TRANSPORT 3 ATTENTION\<0><0><0>
                  00 00 00 4E 4F 49 54 4E 45  0078B
4E 45 54 54 41 20 52 45 54 5F 41 4D 52 4F 46  00794 P.ADT:   .ASCII  \FORMATTER ATTENTION\<0>
                              00 4E 4F 49 54  007A3
4C 49 41 46 20 43 49 54 53 4F 4E 47 41 49 44  007A8 P.ADU:   .ASCII  \DIAGNOSTIC FAILED\<0><0><0>
                              00 00 00 44 45  007B7
4F 43 4F 54 4F 52 50 20 32 20 4C 45 56 45 4C  007BC P.ADV:   .ASCII  \LEVEL 2 PROTOCOL ERROR\<0><0>
                  00 00 52 4F 52 52 45 20 4C  007CB
52 45 20 4E 4F 49 53 53 49 4D 53 4E 41 52 54  007D4 P.ADW:   .ASCII  \TRANSMISSION ERROR\<0><0>
                              00 00 52 4F 52  007E3
52 4F 52 52 45 20 52 45 54 54 41 4D 52 4F 46  007E8 P.ADX:   .ASCII  \FORMATTER ERROR (OPERATIONAL CODE)\<0>
43 20 4C 41 4E 4F 49 54 41 52 45 50 4F 28 20  007F7
                              00 29 45 44 4F  00806
                                              00  0080B           .ASCII  <0>
```

```
                00 00 59 44 41 45 52 20 41 54 41 44 0080C P.ADY:  .ASCII  \DATA READY\<0><0>
                00 45 47 44 45 4C 57 4F 4E 4B 43 41 00818 P.ADZ:  .ASCII  \ACKNOWLEDGE\<0>
                00 00 00 4E 4F 49 54 4E 45 54 54 41 00824 P.AEA:  .ASCII  \ATTENTION\<0><0><0>
49 45 43 45 52 20 52 45 54 54 41 4D 52 4F 46 00830 P.AEB:  .ASCII  \FORMATTER RECEIVER READY\
                                  59 44 41 45 52 20 55 45 56 0083F
00 00 42 20 54 43 45 4C 45 53 20 54 52 4F 50 00848 P.AEC:  .ASCII  \PORT SELECT B\<0><0><0>
                                                 00 00857
00 00 41 20 54 43 45 4C 45 53 20 54 52 4F 50 00858 P.AED:  .ASCII  \PORT SELECT A\<0><0><0>
                                                 00 00867
00 00 44 45 53 53 45 52 50 20 54 4C 55 41 46 00868 P.AEE:  .ASCII  \FAULT PRESSED\<0><0><0>
                                                 00 00877
4F 20 4E 49 20 45 50 41 54 20 44 45 56 4F 4D 00878 P.AEF:  .ASCII  \MOVED TAPE IN OPPOSITE DIRECTION\
49 54 43 45 52 49 44 20 45 54 49 53 4F 50 50 00887
                                              4E 4F 00896
52 20 4E 49 20 45 50 41 54 20 44 45 56 4F 4D 00898 P.AEG:  .ASCII  \MOVED TAPE IN REVERSE DIRECTION\<0>
4F 49 54 43 45 52 49 44 20 45 53 52 45 56 45 008A7
                                           00 4E 008B6
                00 45 54 49 52 57 20 41 20 53 41 57 008B8 P.AEH:  .ASCII  \WAS A WRITE\<0>
20 44 45 44 4E 45 54 58 45 20 38 37 41 54 1E 008C4 P.AEI:  .ASCII  <30>\TA78 EXTENDED FORMATTER STATUS\
55 54 41 54 53 20 52 45 54 54 41 4D 52 4F 46 008D3
                                              53 008E2
                                                 008E3          .BLKB   1
52 45 54 54 41 4D 52 4F 46 20 53 45 4E 49 4C 008E4 P.AEK:  .ASCII  \LINES FORMATTER RECEPTION ENABLED\<0><0>
42 41 4E 45 20 4E 4F 49 54 50 45 43 45 52 20 008F3
                                  00 00 44 45 4C 00902
                                              00 00907          .ASCII  <0>
                                        010E0021  00908 P.AEJ:  .LONG   17694753
                                        00000000' 0090C          .ADDRESS P.AEK
4E 4F 43 20 45 4D 49 54 4C 41 45 52 20 4E 4F 00910 P.AEM:  .ASCII  \ON REALTIME CONTROLLER STATE LINE\<0><0>
4C 20 45 54 41 54 53 20 52 45 4C 4C 4F 52 54 0091F
                                  00 00 45 4E 49 0092E
                                              00 00933          .ASCII  <0>
                                        010E0021  00934 P.AEL:  .LONG   17694753
                                        00000000' 00938          .ADDRESS P.AEM
                                           42 01 0093C P.AEN:  .ASCII  <1>\B\
                                           41 01 0093E P.AEO:  .ASCII  <1>\A\
52 45 20 59 54 49 52 41 50 20 45 43 52 4F 46 00940 P.AEQ:  .ASCII  \FORCE PARITY ERROR\<0><0>
                                  00 00 52 4F 52 0094F
                                        010E0012  00954 P.AEP:  .LONG   17694738
                                        00000000' 00958          .ADDRESS P.AEQ
4F 49 54 41 52 45 50 4F 20 4C 41 4D 52 4F 4E 0095C P.AES:  .ASCII  \NORMAL OPERATION\
                                              4E 00968
                                        010E0010  0096C P.AER:  .LONG   17694736
                                        00000000' 00970          .ADDRESS P.AES
4E 49 4C 4E 4F 20 52 45 54 54 41 4D 52 4F 46 00974 P.AEU:  .ASCII  \FORMATTER ONLINE\
                                              45 00983
                                        010E0010  00984 P.AET:  .LONG   17694736
                                        00000000' 00988          .ADDRESS P.AEU
4C 49 41 56 41 20 52 45 54 54 41 4D 52 4F 46 0098C P.AEW:  .ASCII  \FORMATTER AVAILABLE OR OFFLINE\<0><0>
45 4E 49 4C 46 46 4F 20 52 4F 20 45 4C 42 41 0099B
                                           00 00 009AA
                                        010E001E  009AC P.AEV:  .LONG   17694750
                                        00000000' 009B0          .ADDRESS P.AEW
45 4E 4E 4F 43 20 54 52 4F 50 53 4E 41 52 54 009B4 P.AEY:  .ASCII  \TRANSPORT CONNECTION STATE CHANGE!/!39< \
41 48 43 20 45 54 41 54 53 20 4E 4F 49 54 43 009C3
                                  20 3. 39 33 21 2F 21 45 47 4E 009D2
                                        00 42 5A 21 20 5D 20 54 4E 43 3E 21 009DC          .ASCII  \!>CNT = !ZB\<0>
```

```
                                                                010E0033  009E8 P.AEX:  .LONG   17694771
                                                                00000000' 009EC         .ADDRESS P.AEY
54 20 53 57 4F 4C 4C 41 3E 21 20 3C 39 33 21      009F0 P.AFA:  .ASCII  \!39< !>ALLOWS TRANSMISSION OF STATE!/!39\
20 46 4F 20 4E 4F 49 53 53 49 4D 53 4E 41 52      009FF
                39 33 21 2F 21 45 54 41 54 53     00A0E
41 4E 45 20 4E 4F 20 53 54 49 42 3E 21 20 3C      00A18         .ASCII  \< !>BITS ON ENABLED PORT(S) REALTIME!/!3\
45 52 20 29 53 28 54 52 4F 50 20 44 45 4C 42      00A27
                33 21 2F 21 45 4D 49 54 4C 41     00A36
20 52 45 54 54 41 4D 52 4F 46 3E 21 20 3C 39      00A40         .ASCII  \9< !>FORMATTER STATE LINE\<0><0><0>
         00 00 00 45 4E 49 4C 20 45 54 41 54 53   00A4E
                                                                010E0069  00A5C P.AEZ:  .LONG   17694825
                                                                00000000' 00A60         .ADDRESS P.AFA
52 54 20 45 43 52 4F 46 3E 21 20 3C 39 33 21      00A64 P.AFC:  .ASCII  \!39< !>FORCE TRANSMISSION OF ZEROS ON!/!\
5A 20 46 4F 20 4E 4F 49 53 53 49 4D 53 4E 41      00A73
                21 2F 21 4E 4F 20 53 4F 52 45     00A82
50 20 44 45 4C 42 41 4E 45 3E 21 20 3C 39 33      00A8C         .ASCII  \39< !>ENABLED PORT(S) REALTIME!/!39< !>F\
45 4D 49 54 4C 41 45 52 20 29 53 28 54 52 4F      00A9B
                46 3E 21 20 3C 39 33 21 2F 21     00AAA
20 45 54 41 54 53 20 52 45 54 54 41 4D 52 4F      00AB4         .ASCII  \ORMATTER STATE LINE\<0>
                00 45 4E 49 4C                    00AC3
                                                                010E0063  00AC8 P.AFB:  .LONG   17694819
                                                                00000000' 00ACC         .ADDRESS P.AFC
44 4F 43 20 53 55 54 41 54 53 20 54 53 41 4C      00AD0 P.AFE:  .ASCII  \LAST STATUS CODE = \<0>
                00 20 3D 20 45                    00ADF
                                                                010E0013  00AE4 P.AFD:  .LONG   17694739
                                                                00000000' 00AE8         .ADDRESS P.AFE
                        00 20 3D 20 54 49 4E 55   00AEC P.AFG:  .ASCII  \UNIT = \<0>
                                                                010E0007  00AF4 P.AFF:  .LONG   17694727
                                                                00000000' 00AF8         .ADDRESS P.AFG
44 4E 41 4D 4D 4F 43 20 4C 41 49 54 49 4E 49      00AFC P.AFI:  .ASCII  \INITIAL COMMAND -\<0><0><0>
                00 00 00 2D 20                    00B0B
                                                                010E0011  00B10 P.AFH:  .LONG   17694737
                                                                00000000' 00B14         .ADDRESS P.AFI
53 54 53 45 55 51 45 52 20 59 52 54 45 52 20      00B18 P.AFK:  .ASCII  \ RETRY REQUESTS DURING ERROR\
         52 4F 52 52 45 20 47 4E 49 52 55 44 20   00B27
                                                                010E001C  00B34 P.AFJ:  .LONG   17694748
                                                                00000000' 00B38         .ADDRESS P.AFK
4E 45 55 51 45 53 20 59 52 45 56 4F 43 45 52      00B3C P.AFM:  .ASCII  \RECOVERY SEQUENCE\<0><0><0>
                00 00 00 45 43                    00B4B
                                                                010E0011  00B50 P.AFL:  .LONG   17694737
                                                                00000000' 00B54         .ADDRESS P.AFM
41 54 20 45 43 4E 49 53 20 53 54 49 4E 49 20      00B58 P.AFO:  .ASCII  \ INITS SINCE TA78 MASTER RESET\<0><0>
54 45 53 45 52 20 52 45 54 53 41 4D 20 38 37      00B67
                                        00 00     00B76
                                                                010E001E  00B78 P.AFN:  .LONG   17694750
                                                                00000000' 00B7C         .ADDRESS P.AFO
                  00 00 00 50 55 20 52 57 50 20 52 4F 00B80 P.AFQ:  .ASCII  \OR PWR UP\<0><0><0>
                                                                010E0009  00B8C P.AFP:  .LONG   17694729
                                                                00000000' 00B90         .ADDRESS P.AFQ
44 44 41 20 46 4F 20 53 54 4E 45 54 4E 4F 43      00B94 P.AFS:  .ASCII  \CONTENTS OF ADDRESS IN (SP)\<0>
         00 29 50 53 28 20 4E 49 20 53 53 45 52   00BA3
                                                                010E001B  00BB0 P.AFR:  .LONG   17694747
                                                                00000000' 00BB4         .ADDRESS P.AFS
44 44 41 20 46 4F 20 53 54 4E 45 54 4E 4F 43      00BB8 P.AFU:  .ASCII  \CONTENTS OF ADDRESS IN (SP+1)\<0><0>
00 29 31 2B 50 53 28 20 4E 49 20 53 53 45 52      00BC7
                                        00        00BD6
                                        00        00BD7         .ASCII  <0>
```

```
                                                     010E001D  00BD8 P.AFT:   .LONG    17694749
                                                     00000000' 00BDC          .ADDRESS P.AFU
44 44 41 20 46 4F 20 53 54 4E 45 54 4E 4F 43  00BE0 P.AFW:   .ASCII   \CONTENTS OF ADDRESS IN (SP+2)\<0><0>
00 29 32 2B 50 53 28 20 4E 49 20 53 53 45 52  00BEF
                                          00  00BFE
                                          00  00BFF          .ASCII   <0>
                                                     010E001D  00C00 P.AFV:   .LONG    17694749
                                                     00000000' 00C04          .ADDRESS P.AFW
44 44 41 20 46 4F 20 53 54 4E 45 54 4E 4F 43  00C08 P.AFY:   .ASCII   \CONTENTS OF ADDRESS IN (SP+3)\<0><0>
00 29 33 2B 50 53 28 20 4E 49 20 53 53 45 52  00C17
                                          00  00C26
                                          00  00C27          .ASCII   <0>
                                                     010E001D  00C28 P.AFX:   .LONG    17694749
                                                     00000000' 00C2C          .ADDRESS P.AFY

                                                                              .PSECT   $OWN$,NOEXE,  PIC,2

                                                     0000001E  001CC BYTE4_DESC_TBL:
                                                                              .LONG    30
                                                     00000000' 001D0          .ADDRESS P.ADH
                                                     00000021  001D4          .LONG    33
                                                     00000000' 001D8          .ADDRESS P.ADI
                                                     0000001A  001DC          .LONG    26
                                                     00000000' 001E0          .ADDRESS P.ADJ
                                                     0000001F  001E4          .LONG    31
                                                     00000000' 001E8          .ADDRESS P.ADK
                                                     00000019  001EC          .LONG    25
                                                     00000000' 001F0          .ADDRESS P.ADL
                                                     0000001C  001F4 BYTE5_DESC_TBL:
                                                                              .LONG    28
                                                     00000000' 001F8          .ADDRESS P.ADM
                                                     00000010  001FC          .LONG    16
                                                     00000000' 00200          .ADDRESS P.ADN
                                                     0000001A  00204          .LONG    26
                                                     00000000' 00208          .ADDRESS P.ADO
                                                     00000015  0020C          .LONG    21
                                                     00000000' 00210          .ADDRESS P.ADP
                                                     00000015  00214          .LONG    21
                                                     00000000' 00218          .ADDRESS P.ADQ
                                                     00000015  0021C          .LONG    21
                                                     00000000' 00220          .ADDRESS P.ADR
                                                     00000015  00224          .LONG    21
                                                     00000000' 00228          .ADDRESS P.ADS
                                                     00000013  0022C          .LONG    19
                                                     00000000' 00230          .ADDRESS P.ADT
                                                     00000011  00234 BYTE6_DESC_TBL:
                                                                              .LONG    17
                                                     00000000' 00238          .ADDRESS P.ADU
                                                     00000016  0023C          .LONG    22
                                                     00000000' 00240          .ADDRESS P.ADV
                                                     00000012  00244          .LONG    18
                                                     00000000' 00248          .ADDRESS P.ADW
                                                     00000022  0024C          .LONG    34
                                                     00000000' 00250          .ADDRESS P.ADX
                                                     0000000A  00254 BYTE10_DESC_TBL:
                                                                              .LONG    10
                                                     00000000' 00258          .ADDRESS P.ADY
```

```
                00000008  0025C          .LONG   11
                00000000' 00260          .ADDRESS P.ADZ
                00000009  00264          .LONG   9
                00000000' 00268          .ADDRESS P.AEA
                00000018  0026C          .LONG   24
                00000000' 00270          .ADDRESS P.AEB
                0000000D  00274 BYTE11_DESC_TBL:
                                         .LONG   13
                00000000' 00278          .ADDRESS P.AEC
                0000000D  0027C          .LONG   13
                00000000' 00280          .ADDRESS P.AED
                0000000D  00284          .LONG   13
                00000000' 00288          .ADDRESS P.AEE
                00000020  0028C BYTE13_DESC_TBL:
                                         .LONG   32
                00000000' 00290          .ADDRESS P.AEF
                0000001F  00294          .LONG   31
                00000000' 00298          .ADDRESS P.AEG
                0000000B  0029C          .LONG   11
                00000000' 002A0          .ADDRESS P.AEH



                                         .PSECT  $CODE,NOWRT,  PIC,2

                      00FC 00000         .ENTRY  TA78_FORMATTER_STS_DECODE, Save R2,R3,R4,-  ; 0833
                                                 R5,R6,R7
            57 00000000G   00  9E 00002  MOVAB   TRANSLATE_BITS, R7
            56 00000000V   00  9E 00009  MOVAB   LABEL_AND_HEX_OUTPUT, R6
            55 00000000G   00  9E 00010  MOVAB   OUTPUT_LINES, R5
            54 00000000'   00  9E 00017  MOVAB   P.AEI, R4
            53 00000000G   00  9E 0001E  MOVAB   FRMTR_STS+8, R3
            52 00000000'   00  9E 00025  MOVAB   ARGLIST, R2
            5E             04  C2 0002C  SUBL2   #4, SP
            62             64  9E 0002F  MOVAB   P.AEI, ARGLIST                          0905
         00A0  C2                01  90 00032  MOVB    #1, MAIN_HDR                      0906
                                 52  DD 00037  PUSHL   R2                                0907
                              68 A2  DD 00039  PUSHL   FAO_STRINGS
                            65   02  FB 0003C  CALLS   #2, OUTPUT_LINES
                                 02  DD 0003F  PUSHL   #2                                0909
                                 01  DD 00041  PUSHL   #1
                            06   02  FB 00043  CALLS   #2, LABEL_AND_HEX_OUTPUT
                            7E     F8 A3  3C 00046  MOVZWL  ERRNUM, -(SP)                 0911
         00000000V          00   01  FB 0004A  CALLS   #1, ERROR_NUMBER_BYTE_DECODE
                                 03  DD 00051  PUSHL   #3                                0914
                            66   01  FB 00053  CALLS   #1, LABEL_AND_HEX_OUTPUT
                            62 00A8  C2  DD 00056  MOVL    STI_STRING, ARGLIST           0917
               04           A2     FA A3  9A 0005B  MOVZBL  FRMTR_STS+2, ARGLIST+4       0918
                                 52  DD 00060  PUSHL   R2                                0920
                          0084   C2  DD 00062  PUSHL   FAO_STRINGS+28
                            65   02  FB 00066  CALLS   #2, OUTPUT_LINES
                                 04  DD 00069  PUSHL   #4                                0925
                            66   01  FB 0006B  CALLS   #1, LABEL_AND_HEX_OUTPUT
                                 6E  D4 0006E  CLRL    (SP)                              0928
                                 5E  DD 00070  PUSHL   SP
                              64 A2  9F 00072  PUSHAB  FAO_STRING                        0927
                          00A4   C2  9F 00075  PUSHAB  OUT_ARGLIST
```

```
                              01CC  C2 9F 00079        PUSHAB  BYTE4_DESC_TBL                          :
                                44  A2 9F 0007D        PUSHAB  BIT_MSK_5                               :
                                FB  A3 9F 00080        PUSHAB  FRMTR_STS+3                             :
                      67            06 FB 00083        CALLS   #6, TRANSLATE_BITS                      :
                      0A            50 E9 00086        BLBC    R0, 1$                                  :
                              00A4  C2 DD 00089        PUSHL   OUT_ARGLIST                             : 0930
                                64  A2 DD 0008D        PUSHL   FAO_STRING                              :
                      65            02 FB 00090        CALLS   #2, OUTPUT_LINES                        :
                                    05 DD 00093 1$:    PUSHL   #5                                      : 0935
                      66            01 FB 00095        CALLS   #1, LABEL_AND_HEX_OUTPUT                :
                                    6E D4 00098        CLRL    (SP)                                    : 0938
                                    5E DD 0009A        PUSHL   SP                                      :
                                64  A2 9F 0009C        PUSHAB  FAO_STRING                              : 0937
                              00A4  C2 9F 0009F        PUSHAB  OUT_ARGLIST                             :
                              01F4  C2 9F 000A3        PUSHAB  BYTE5_DESC_TBL                          :
                                30  A2 9F 000A7        PUSHAB  BIT_MSK_0                               :
                                FC  A3 9F 000AA        PUSHAB  FRMTR_STS+4                             :
                      67            06 FB 000AD        CALLS   #6, TRANSLATE_BITS                      :
                      0A            50 E9 000B0        BLBC    R0, 2$                                  :
                              (CA4  C2 DD 000B3        PUSHL   OUT_ARGLIST                             : 0940
                                64  A2 DD 000B7        PUSHL   FAO_STRING                              :
                      65            02 FB 000BA        CALLS   #2, OUTPUT_LINES                        :
                                    06 DD 000BD 2$:    PUSHL   #6                                      : 0946
                      66            01 FB 000BF        CALLS   #1, LABEL_AND_HEX_OUTPUT                :
                                    6E D4 000C2        CLRL    (SP)                                    : 0949
                                    5E DD 000C4        PUSHL   SP                                      :
                                64  A2 9F 000C6        PUSHAB  FAO_STRING                              : 0948
                              00A4  C2 9F 000C9        PUSHAB  OUT_ARGLIST                             :
                              0234  C2 9F 000CD        PUSHAB  BYTE6_DESC_TBL                          :
                                34  A2 9F 000D1        PUSHAB  BIT_MSK_1                               :
                                FD  A3 9F 000D4        PUSHAB  FRMTR_STS+5                             :
                      67            06 FB 000D7        CALLS   #6, TRANSLATE_BITS                      :
                      0A            50 E9 000DA        BLBC    R0, 3$                                  :
                              00A4  C2 DD 000DD        PUSHL   OUT_ARGLIST                             : 0951
                                64  A2 DD 000E1        PUSHL   FAO_STRING                              :
                      65            02 FB 000E4        CALLS   #2, OUTPUT_LINES                        :
                                    07 DD 000E7 3$:    PUSHL   #7                                      : 0957
                      66            01 FB 000E9        CALLS   #1, LABEL_AND_HEX_OUTPUT                :
                      7E        FE  A3 9A 000EC        MOVZBL  FRMTR_STS+6, -(SP)                      : 0959
                FE60  CF            01 FB 000F0        CALLS   #1, SUMMARY_MODE_BYTE_DECODE            :
                                    08 DD 000F5        PUSHL   #8                                      : 0965
                      66            01 FB 000F7        CALLS   #1, LABEL_AND_HEX_OUTPUT                :
                      7E        FF  A3 9A 000FA        MOVZBL  FRMTR_STS+7, -(SP)                      : 0967
                FEC8  CF            01 FB 000FE        CALLS   #1, CNTRLR_FLG_BYTE_DECODE              :
                                    09 DD 00103        PUSHL   #9                                      : 0972
                      66            01 FB 00105        CALLS   #1, LABEL_AND_HEX_OUTPUT                :
                      51            63 D0 00108        MOVL    FRMTR_STS+8, R1                         : 0974
                      04            51 E8 0010B        BLBS    R1, 4$                                  :
          0F          63            01 E1 0010E        BBC     #1, FRMTR_STS+8, 5$                     : 0975
                 10   A2        44  A4 9E 00112 4$:    MOVAB   P.AEJ, ARGLIST+16                       : 0978
                 14   A2        70  A4 9E 00117        MOVAB   P.AEL, ARGLIST+20                       : 0979
                 64   A2        7C  A2 D0 0011C        MOVL    FAO_STRINGS+20, FAO_STRING              : 0980
  50      63          20            02 EF 00121 5$:    EXTZV   #2, #32, FRMTR_STS+8, R0                : 0983
                      04            50 E8 00126        BLBS    R0, 6$                                  :
          06          63            03 E1 00129        BBC     #3, FRMTR_STS+8, 7$                     : 0984
                 64   A2      0080  C2 D0 0012D 6$:    MOVL    FAO_STRINGS+24, FAO_STRING              : 0986
                      03            51 E8 00133 7$:    BLBS    R1, 8$                                  : 0988
```

```
                        05              50  E9 00136          BLBC    R0, 9$                                          ; 0989
                    0C  A2          78  A4  9E 00139 8$:      MOVAB   P.AEN, ARGLIST+12                               ; 0991
                04      63              01  E0 0013E 9$:      BBS     #1, FRMTR_STS+8, 10$                            ; 0993
                05      63              03  E1 00142          BBC     #3, FRMTR_STS+8, 11$                            ; 0994
                    0C  A2          7A  A4  9E 00146 10$:     MOVAB   P.AEO, ARGLIST+12                               ; 0996
                                    0C  A2  9F 0014B 11$:     PUSHAB  ARGLIST+12                                      ; 0999
                                    64  A2  DD 0014E          PUSHL   FAO_STRING
                                        65  02  FB 00151      CALLS   #2, OUTPUT_LINES
                                            0A  DD 00154      PUSHL   #10                                             ; 1005
                                        66  01  FB 00156      CALLS   #1, LABEL_AND_HEX_OUTPUT
                                        07      01  A3  E9 00159  BLBC    FRMTR_STS+9, 12$                            ; 1007
                                        62  0090    C4  9E 0015D  MOVAB   P.AEP, ARGLIST                              ; 1009
                                            05  11 00162      BRB     13$
                                        62  00A8    C4  9E 00164 12$:     MOVAB   P.AER, ARGLIST                      ; 1011
                08      01  A3              01  E0 00169 13$:     BBS     #1, FRMTR_STS+9, 14$                        ; 1013
                        04  A2      00C0    C4  9E 0016E          MOVAB   P.AET, ARGLIST+4                            ; 1015
                                            06  11 00174      BRB     15$
                        04  A2      00E8    C4  9E 00176 14$:     MOVAB   P.AEV, ARGLIST+4                            ; 1017
                                    32      FB  A3  E9 0017C 15$:     BLBC    FRMTR_STS+3, 16$                        ; 1020
                                            6E  D4 00180      CLRL    (SP)                                            ; 1024
                                            5E  DD 00182      PUSHL   SP
                                        64  A2  9F 00184      PUSHAB  FAO_STRING                                      ; 1023
                                    00A4    C2  9F 00187      PUSHAB  OUT_ARGLIST
                                    0254    C2  9F 0018B      PUSHAB  BYTE10_DESC_TBL
                                        48  A2  9F 0018F      PUSHAB  BIT_MSR_6
                                        01  A3  9F 00192      PUSHAB  FRMTR_STS+9
                                        67      06  FB 00195  CALLS   #6, TRANSLATE_BITS
                                        51      50  E9 00198  BLBC    R0, 17$
                                    00A4    C2  DD 0019B      PUSHL   OUT_ARGLIST                                     ; 1028
                                        64  A2  DD 0019F      PUSHL   FAO_STRING
                                        04  A2  9F 001A2      PUSHAB  ARGLIST+4                                       ; 1027
                                        50  74  A2  DO 001A5  MOVL    FAO_STRINGS+12, R0
                                            50  DD 001A9      PUSHL   R0                                              ; 1028
                                            05  BB 001AB      PUSHR   #^M<R0,R2>
                                        65      06  FB 001AD  CALLS   #6, OUTPUT_LINES
                                            3A  11 001B0      BRB     17$                                             ; 1020
                        35      FB  A3          01  E1 001B2 16$:     BBC     #1, FRMTR_STS+3, 17$                    ; 1032
                                64  A2      0124    C4  9E 001B7  MOVAB   P.AEX, FAO_STRING                           ; 1035
0C  A2          01  A3              05              02  EF 001BD  EXTZV   #2, #5, FRMTR_STS+9, ARGLIST+12            ; 1036
                                            0C  A2  9F 001C4      PUSHAB  ARGLIST+12                                  ; 1040
                                            64  A2  DD 001C7      PUSHL   FAO_STRING
                                            04  A2  9F 001CA      PUSHAB  ARGLIST+4                                   ; 1039
                                        50  74  A2  DO 001CD  MOVL    FAO_STRINGS+12, R0
                                            50  DD 001D1      PUSHL   R0
                                            05  BB 001D3      PUSHR   #^M<R0,R2>                                      ; 103?
                                        65      06  FB 001D5  CALLS   #6, OUTPUT_LINES
                                    0F      01  A3          06  E1 001D8  BBC     #6, FRMTR_STS+9, 17$                ; 1042
                                    0C  A2      0284    C2  9E 001DD  MOVAB   BYTE10_DESC_TBL+48, ARGLIST+12          ; 1045
                                            0C  A2  9F 001E3      PUSHAB  ARGLIST+12                                  ; 1046
                                            74  A2  DD 001E6      PUSHL   FAO_STRINGS+12
                                        65      02  FB 001E9  CALLS   #2, OUTPUT_LINES
                                            01  A3  95 001EC 17$:     TSTB    FRMTR_STS+9                             ; 1052
                                            08  18 001EF      BGEQ    18$
                                        64  A2      0198    C4  9E 001F1  MOVAB   P.AEZ, FAO_STRING                   ; 1054
                                            06  11 001F7      BRB     19$
                                        64  A2      0204    C4  9E 001F9 18$:     MOVAB   P.AFB, FAO_STRING           ; 1056
                                            6E  D4 001FF 19$:     CLRL    (SP)                                        ; 1058
```

```
                                  5E  DD 00201          PUSHL    SP                                    :
                            64    A2  DD 00203          PUSHL    FAO_STRING                            :
                    65            02  FB 00206          CALLS    #2, OUTPUT_LINES                      :
                                  0B  DD 00209          PUSHL    #11                                   : 1063
                    66            01  FB 0020B          CALLS    #1, LABEL_AND_HEX_OUTPUT              :
                                  6E  D4 0020E          CLRL     (SP)                                  : 1066
                                  5E  DD 00210          PUSHL    SP                                    :
                            64    A2  9F 00212          PUSHAB   FAO_STRING                            : 1065
                          00A4    C2  9F 00215          PUSHAB   OUT_ARGLIST                           :
                          0274    C2  9F 00219          PUSHAB   BYTE11_DESC_TBL                       :
                            40    A2  9F 0021D          PUSHAB   BIT_MSK_4                             :
                            02    A3  9F 00220          PUSHAB   FRMTR_STS+10                          :
                    67            06  FB 00223          CALLS    #6, TRANSLATE_BITS                    :
                    0A            50  E9 00226          BLBC     R0, 20$                               :
                          00A4    C2  DD 00229          PUSHL    OUT_ARGLIST                           : 1068
                            64    A2  DD 0022D          PUSHL    FAO_STRING                            :
                    65            02  FB 00230          CALLS    #2, OUTPUT_LINES                      :
                                  0C  DD 00233 20$:     PUSHL    #12                                   : 1074
                    66            01  FB 00235          CALLS    #1, LABEL_AND_HEX_OUTPUT              :
                    62    0220    C4  9E 00238          MOVAB    P.AFD, ARGLIST                        : 1076
               04   A2      03    A3  9A 0023D          MOVZBL   FRMTR_STS+11, ARGLIST+4               : 1077
                                  52  DD 00242          PUSHL    R2                                    : 1079
                          0084    C2  DD 00244          PUSHL    FAO_STRINGS+28                        :
                    65            02  FB 00248          CALLS    #2, OUTPUT_LINES                      :
                                  0D  DD 0024B          PUSHL    #13                                   : 1084
                    66            01  FB 0024D          CALLS    #1, LABEL_AND_HEX_OUTPUT              :
                    62    0230    C4  9E 00250          MOVAB    P.AFF, ARGLIST                        : 1086
04   A2   04   A3           02    01  EF 00255          EXTZV    #1, #2, FRMTR_STS+12, ARGLIST+4       : 1087
                                  52  DD 0025C          PUSHL    R2                                    : 1089
                            70    A2  DD 0025E          PUSHL    FAO_STRINGS+8                         :
                    65            02  FB 00261          CALLS    #2, OUTPUT_LINES                      :
               F8   8F      04    A3  93 00264          BITB     FRMTR_STS+12, #248                    : 1091
                                  34  13 00269          BEQL     21$                                   :
               08   A2    024C    C4  9E 0026B          MOVAB    P.AFH, ARGLIST+8                      : 1094
                            08    A2  9F 00271          PUSHAB   ARGLIST+8                             : 1095
                            74    A2  DD 00274          PUSHL    FAO_STRINGS+12                        :
                    65            02  FB 00277          CALLS    #2, OUTPUT_LINES                      :
                                  6E  D4 0027A          CLRL     (SP)                                  : 1098
                                  5E  DD 0027C          PUSHL    SP                                    :
                            64    A2  9F 0027E          PUSHAB   FAO_STRING                            : 1097
                          00A4    C2  9F 00281          PUSHAB   OUT_ARGLIST                           :
                          028C    C2  9F 00285          PUSHAB   BYTE13_DESC_TBL                       :
                            4C    A2  9F 00289          PUSHAB   BIT_MSK_7                             :
                            04    A3  9F 0028C          PUSHAB   FRMTR_STS+12                          :
                    67            06  FB 0028F          CALLS    #6, TRANSLATE_BITS                    :
                    0A            50  E9 00292          BLBC     R0, 21$                               :
                          00A4    C2  DD 00295          PUSHL    OUT_ARGLIST                           : 1100
                            64    A2  DD 00299          PUSHL    FAO_STRING                            :
                    65            02  FB 0029C          CALLS    #2, OUTPUT_LINES                      :
                                  0E  DD 0029F 21$:     PUSHL    #14                                   : 1106
                    66            01  FB 002A1          CALLS    #1, LABEL_AND_HEX_OUTPUT              :
                    62      05    A3  9A 002A4          MOVZBL   FRMTR_STS+13, ARGLIST                 : 1108
               04   A2    0270    C4  9E 002A8          MOVAB    P.AFJ, ARGLIST+4                      : 1109
               08   A2    028C    C4  9E 002AE          MOVAB    P.AFL, ARGLIST+8                      : 1110
                                  52  DD 002B4          PUSHL    R2                                    : 1112
                          0088    C2  DD 002B6          PUSHL    FAO_STRINGS+32                        :
                    65            02  FB 002BA          CALLS    #2, OUTPUT_LINES                      :
```

```
                                 0F   DD 002BD          PUSHL   #15                                    ; 1117
                           66    01   FB 002BF          CALLS   #1, LABEL_AND_HEX_OUTPUT
                           62    06   A3 9A 002C2       MOVZBL  FRMTR_STS+14, ARGLIST                  ; 1119
                     04    A2  02B4   C4 9E 002C6       MOVAB   P.AFN, ARGLIST+4                       ; 1120
                     08    A2  02C8   C4 9E 002CC       MOVAB   P.AFP, ARGLIST+8                       ; 1121
                                      52 DD 002D2       PUSHL   R2                                     ; 1123
                           0088       C2 DD 002D4       PUSHL   FAO_STRINGS+32
                           65         02 FB 002D8       CALLS   #2, OUTPUT_LINES
                                      10 DD 002DB       PUSHL   #16                                    ; 1128
                           66         01 FB 002DD       CALLS   #1, LABEL_AND_HEX_OUTPUT
                 3C5F      8F    F8   A3 B1 002E0       CMPW    ERRNUM, #T5455                         ; 1130
                                      0D 12 002E6       BNEQ    22$
                           62  02EC   C4 9E 002E8       MOVAB   P.AFR, ARGLIST                         ; 1133
                                      52 DD 002ED       PUSHL   R2                                     ; 1135
                              74      A2 DD 002EF       PUSHL   FAO_STRINGS+12
                           65         02 FB 002F2       CALLS   #2, OUTPUT_LINES
                                      11 DD 002F5 22$:  PUSHL   #17                                    ; 1142
                           66         01 FB 002F7       CALLS   #1, LABEL_AND_HEX_OUTPUT
                 3C5F      8F    F8   A3 B1 002FA       CMPW    ERRNUM, #T5455                         ; 1144
                                      0D 12 00300       BNEQ    23$
                           62  0314   C4 9E 00302       MOVAB   P.AFT, ARGLIST                         ; 1147
                                      52 DD 00307       PUSHL   R2                                     ; 1149
                              74      A2 DD 00309       PUSHL   FAO_STRINGS+12
                           65         02 FB 0030C       CALLS   #2, OUTPUT_LINES
                                      12 DD 0030F 23$:  PUSHL   #18                                    ; 1156
                           66         01 FB 00311       CALLS   #1, LABEL_AND_HEX_OUTPUT
                 3C5F      8F    F8   A3 B1 00314       CMPW    ERRNUM, #T5455                         ; 1158
                                      0D 12 0031A       BNEQ    24$
                           62  033C   C4 9E 0031C       MOVAB   P.AFV, ARGLIST                         ; 1161
                                      52 DD 00321       PUSHL   R2                                     ; 1163
                              74      A2 DD 00323       PUSHL   FAO_STRINGS+12
                           65         02 FB 00326       CALLS   #2, OUTPUT_LINES
                                      13 DD 00329 24$:  PUSHL   #19                                    ; 1170
                           66         01 FB 0032B       CALLS   #1, LABEL_AND_HEX_OUTPUT
                 3C5F      8F    F8   A3 B1 0032E       CMPW    ERRNUM, #T5455                         ; 1172
                                      0D 12 00334       BNEQ    25$
                           62  0364   C4 9E 00336       MOVAB   P.AFX, ARGLIST                         ; 1175
                                      52 DD 00338       PUSHL   R2                                     ; 1177
                              74      A2 DD 0033D       PUSHL   FAO_STRINGS+12
                           65         02 FB 00340       CALLS   #2, OUTPUT_LINES
                                      14 DD 00343 25$:  PUSHL   #20                                    ; 1184
                           66         01 FB 00345       CALLS   #1, LABEL_AND_HEX_OUTPUT
                                      04 00348          RET                                            ; 1186

; Routine Size:  841 bytes,    Routine Base:  $CODE + 01B2

;  776          1187  1
```

```
;  778          1188  1 Routine ERROR_NUMBER_BYTE_DECODE (error_number) : NOVALUE =
;  779          1189  2 Begin
;  780          1190  2 !++
;  781          1191  2 !
;  782          1192  2 ! Functional Description:
;  783          1193  2 !
;  784          1194  2 !       This routine decodes and outputs the error number bytes. It is called
;  785          1195  2 !       by the TA78_formatter_sts_decode and the TA78_drive_sts_decode routines.
;  786          1196  2 !
;  787          1197  2 ! Calling Sequence:
;  788          1198  2 !
;  789          1199  2 !       ERROR_NUMBER_BYTE_DECODE (error_number) ;
;  790          1200  2 !
;  791          1201  2 ! Input Parameters:
;  792          1202  2 !
;  793          1203  2 !       Error_number = contents of the error number bytes.
;  794          1204  2 !
;  795          1205  2 ! Output Parameters:
;  796          1206  2 !
;  797          1207  2 !       None.
;  798          1208  2 !
;  799          1209  2 !--
;  800          1210  2
;  801        P 1211  2 STORE_STRINGS ( byte1,
;  802        P 1212  2                 'DIAGNOSTIC DATA AVAILABLE',
;  803          1213  2                 'CRITICAL ERROR' ) ;
;  804          1214  2
;  805          1215  2
;  806          1216  2 Arglist[0] = %ASCID 'ERROR ID NUMBER = ' ;
;  807          1217  2 Arglist[1] = .error_number<0,10> ;
;  808          1218  2
;  809          1219  2 If .error_number<2>
;  810          1220  2 Then
;  811          1221  2     Arglist[2] = %ASCID 'OPERATIONAL ERROR'
;  812          1222  2 Else
;  813          1223  2     Arglist[2] = %ASCID 'DIAGNOSTIC ERROR' ;
;  814          1224  2
;  815          1225  2 Case .error_number<11,3> from 1 TO 7 of
;  816          1226  2     Set
;  817          1227  2     [1]: Arglist[3] = %ASCID 'MISCELLANEOUS' ;
;  818          1228  2     [2]: Arglist[3] = %ASCID 'READ' ;
;  819          1229  2     [3]: Arglist[3] = %ASCID 'WRITE' ;
;  820          1230  2     [4]: Arglist[3] = %ASCID 'TU PORT' ;
;  821          1231  2     [5]: Arglist[3] = %ASCID 'ERROR CORRECTION' ;
;  822          1232  2     [6]: Arglist[3] = %ASCID 'STI COMMUNICATION' ;
;  823          1233  2     [7]: Arglist[3] = %ASCID 'MICROCOMPUTER' ;
;  824          1234  2
;  825          1235  2     [OUTRANGE]: Arglist[3] = %ASCID 'UNKNOWN FAULT NUMBER' ;
;  826          1236  2     Tes ;
;  827          1237  2
;  828          1238  2 OUTPUT_LINES (  .fao_strings[2], arglist[0],
;  829          1239  2                 .fao_strings[3], arglist[2],
;  830          1240  2                 .fao_strings[4], arglist[3] ) ;
;  831          1241  2
;  832          1242  3 If ( TRANSLATE_BITS (error_number,bit_msk_3,byte1_desc_tbl,
;  833          1243  3                       out_arglist,fao_string,%REF(0)) )
;  834          1244  2 Then
```

```
;  835          1245  2     OUTPUT_LINES ( .fao_string, out_arglist[0] ) ;
;  836          1246  2
;  837          1247  2 Return ;
;  838          1248  1 End ;   ! Routine



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

41 54 41 44 20 43 49 54 53 4F 4E 47 41 49 44  00C30 P.AFZ:  .ASCII  \DIAGNOSTIC DATA AVAI: ABLE\<0><0><0>
      00 00 00 45 4C 42 41 4C 49 41 56 41 20  00C3F
00 52 4F 52 52 45 20 4C 41 43 49 54 49 52 43  00C4C P.AGA:  .ASCII  \CRITICAL ERROR\<0><0>
                                           00  00C5B
52 45 42 4D 55 4E 20 44 49 20 52 4F 52 52 45  00C5C P.AGC:  .ASCII  \ERROR ID NUMBER = \<0><0>
                              00 00 20 3D 20  00C6B
                                 010E0012     00C70 P.AGB:  .LONG   17694738
                                 00000000'    00C74         .ADDRESS P.AGC
52 52 45 20 4C 41 4E 4F 49 54 41 52 45 50 4F  00C78 P.AGE:  .ASCII  \OPERATIONAL ERROR\<0><0><0>
                              00 00 00 52 4F  00C87
                                 010E0011     00C8C P.AGD:  .LONG   17694737
                                 00000000'    00C90         .ADDRESS P.AGE
4F 52 52 45 20 43 49 54 53 4F 4E 47 41 49 44  00C94 P.AGG:  .ASCII  \DIAGNOSTIC ERROR\
                                           52  00CA3
                                 010E0010     00CA4 P.AGF:  .LONG   17694 36
                                 00000000'    00CA8         .ADDRESS P.AGG
00 00 53 55 4F 45 4E 41 4C 4C 45 43 53 49 4D  00CAC P.AGI:  .ASCII  \MISCELLANEOUS\<0><0><0>
                                           00  00CBB
                                 010E000D     00CBC P.AGH:  .LONG   17694733
                                 00000000'    00CC0         .ADDRESS P.AGI
                              44 41 45 52     00CC4 P.AGK:  .ASCII  \READ\
                                 010E0004     00CC8 P.AGJ:  .LONG   17694724
                                 00000000'    00CCC         .ADDRESS P.AGK
                           00 00 00 45 54 49 52 57  00CD0 P.AGM:  .ASCII  \WRITE\<0><0><0>
                                 010E0005     00CD8 P.AGL:  .LONG   17694725
                                 00000000'    00CDC         .ADDRESS P.AGM
                           00 54 52 4F 50 20 55 54  00CE0 P.AGO:  .ASCII  \TU PORT\<0>
                                 010E0007     00CE8 P.AGN:  .LONG   17694727
                                 00000000'    00CEC         .ADDRESS P.AGO
4F 49 54 43 45 52 52 4F 43 20 52 4F 52 52 45  00CF0 P.AGQ:  .ASCII  \ERROR CORRECTION\
                                           4E  00CFF
                                 010E0010     00D00 P.AGP:  .LONG   17694736
                                 00000000'    00D04         .ADDRESS P.AGQ
49 54 41 43 49 4E 55 4D 4D 4F 43 20 49 54 53  00D08 P.AGS:  .ASCII  \STI COMMUNICATION\<0><0><0>
                              00 00 00 4E 4F  00D17
                                 010E0011     00D1C P.AGR:  .LONG   17694737
                                 00000000'    00D20         .ADDRESS P.AGS
00 00 52 45 54 55 50 4D 4F 43 4F 52 43 49 4D  00D24 P.AGU:  .ASCII  \MICROCOMPUTER\<0><0><0>
                                           00  00D33
                                 010E000D     00D34 P.AGT:  .LONG   17694733
                                 00000000'    00D38         .ADDRESS P.AGU
4E 20 54 4C 55 41 46 20 4E 57 4F 4E 48 4E 55  00D3C P.AGW:  .ASCII  \UNKNOWN FAULT NUMBER\
                              52 45 42 4D 55  00D4B
                                 010E0014     00D50 P.AGV:  .LONG   17694740
                                 00000000'    00D54         .ADDRESS P.AGW

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

```
                                00000019  002A4 BYTE1_DESC_TBL:
                                                .LONG   25
                                00000000' 002A8         .ADDRESS P.AFZ
                                0000000E  002AC         .LONG   14
                                00C00000' 002B0         .ADDRESS P.AGA



                                                .PSECT  $CODE,NOWRT,  PIC,2

                                          001C 00000 ERROR_NUMBER_BYTE_DECODE:
                                                .WORD   Save R2,R3,R4                                   ; 1188
                                    54 00000000G  00  9E 00002          MOVAB   OUTPUT_LINES, R4
                                    53 00000000'  00  9E 00009          MOVAB   P.AGB, R3
                                    52 00000000'  00  9E 00010          MOVAB   ARGLIST+12, R2
                                    5E            04  C2 00017          SUBL2   #4, SP
                                F4  A2            63  9E 0001A          MOVAB   P.AGB, ARGLIST                                ; 1216
    F8  A2      04  AC            0A              00  EF 0001E          EXTZV   #0, #10, ERROR_NUMBER, ARGLIST+4              ; 1217
                    07        04  AC              02  E1 00025          BBC     #2, ERROR_NUMBER, 1$                          ; 1219
                              FC  A2      1C      A3  9E 0002A          MOVAB   P.AGD, ARGLIST+8                              ; 1221
                                                05  11 0002F          BRB     2$
                              FC  A2      34      A3  9E 00031  1$:     MOVAB   P.AGF, ARGLIST+8                              ; 1223
        50      05  AC                    03      03  EF 00036  2$:     EXTZV   #3, #3, ERROR_NUMBER+1, R0                    ; 1225
                    06                    01      50  CF 0003C          CASEL   R0, #1, #6
      0027          0021              001B        0015      00040  3$.    .WORD   4$-3$,-
                    0038              0034        002D      00048                  5$-3$,-
                                                                                   6$-3$,-
                                                                                   7$-3$,-
                                                                                   8$-3$,-
                                                                                   9$-3$,-
                                                                                   10$-3$
    62      00E0        C3  9E 0004E          MOVAB   P.AGV, ARGLIST+12                                                  ; 1235
                        2B  11 00053          BRB     11$
    62        4C        A3  9E 00055  4$:     MOVAB   P.AGH, ARGLIST+12                                                 ; 1227
                        25  11 00059          BRB     11$
    62        58        A3  9E 0005B  5$:     MOVAB   P.AGJ, ARGLIST+12                                                 ; 1228
                        1F  11 0005F          BRB     11$
    62        68        A3  9E 00061  6$:     MOVAB   P.AGL, ARGLIST+12                                                 ; 1229
                        19  11 00065          BRB     11$
    62        78        A3  9E 00067  7$:     MOVAB   P.AGN, ARGLIST+12                                                 ; 1230
                        13  11 0006B          BRB     11$
    62      0090        C3  9E 0006D  8$:     MOVAB   P.AGP, ARGLIST+12                                                 ; 1231
                        0C  11 00072          BRB     11$
    62      00AC        C3  9E 00074  9$:     MOVAB   P.AGR, ARGLIST+12                                                 ; 1232
                        05  11 00079          BRB     11$
    62      00C4        C3  9E 0007B  10$:    MOVAB   P.AGT, ARGLIST+12                                                 ; 1233
                        52  DD 00080  11$:    PUSHL   R2                                                                ; 1240
                    6C  A2  DD 00082          PUSHL   FAO_STRINGS+16
                    FC  A2  9F 00085          PUSHAB  ARGLIST+8                                                         ; 1239
                    68  A2  DD 00088          PUSHL   FAO_STRINGS+12
                    F4  A2  9F 0008B          PUSHAB  ARGLIST                                                           ; 1238
                    64  A2  DD 0008E          PUSHL   FAO_STRINGS+8
    64                  06  FB 00091          CALLS   #6, OUTPUT_LINES
                        6E  D4 00094          CLRL    (SP)                                                              ; 1243
                        5E  DD 00096          PUSHL   SP
                    58  A2  9F 00098          PUSHAB  FAO_STRING                                                        ; 1242
                  0098  C2  9F 0009B          PUSHAB  OUT_ARGLIST
```

```
                                0298  C2  9F 0009F          PUSHAB  BYTE1_DESC_TBL                    ;
                                  30  A2  9F 000A3          PUSHAB  BIT_MSK_3                        ;
                                  04  AC  9F 000A6          PUSHAB  ERROR_NUMBER                     ;
                00000000G  00         06  FB 000A9          CALLS   #6, TRANSLATE_BITS               ;
                           0A         50  E9 000B0          BLBC    R0, 12$                          ;
                                0098  C2  DD 000B3          PUSHL   OUT_ARGLIST                      ; 1245
                                  58  A2  DD 000B7          PUSHL   FAO_STRING                       ;
                           64         02  FB 000BA          CALLS   #2, OUTPUT_LINES                 ;
                                      04 000BD 12$:         RET                                      ; 1248

; Routine Size:  190 bytes,    Routine Base:  $CODE + 04FB

;  839          1249  1
```

```
;   841          1250  1 GLOBAL Routine TA78_DRIVE_STS_DECODE : NOVALUE -
;   842          1251  2 Begin
;   843          1252  2
;   844          1253  2 !++
;   845          1254  2 !
;   846          1255  2 ! Functional Description:
;   847          1256  2 !
;   848          1257  2 !       This routine decodes and outputs the extended drive status information
;   849          1258  2 !       that is appended to a logmessage 'STI drive error' log entry for a TA78.
;   850          1259  2 !
;   851          1260  2 ! Calling Sequence:
;   852          1261  2 !
;   853          1262  2 !       TA78_DRIVE_STS_DECODE () ;
;   854          1263  2 !
;   855          1264  2 ! Input Parameters:
;   856          1265  2 !
;   857          1266  2 !       None.
;   858          1267  2 !
;   859          1268  2 ! Output Parameters:
;   860          1269  2 !
;   861          1270  2 !       None.
;   862          1271  2 !
;   863          1272  2 !--
;   864          1273  2 Own
;   865          1274  2     B1_diag_sts_codes:  VECTOR [5,byte]
;   866          1275  2                         Initial (byte (7,%X'E',%X'16',%X'2D',%X'35') ),
;   867          1276  2     B2_diag_sts_codes:  VECTOR [7,byte]
;   868          1277  2                         Initial (byte (1,2,3,4,8,%X'D',%X'15') ),
;   869          1278  2     Rmc_sts_codes:      VECTOR [21,byte]
;   870          1279  2                         Initial (byte (1,%X'41',%X'42',%X'43',%X'44',%X'46',
;   871          1280  2                                 %X'81',%X'88',%X'89',%X'8A',%X'90',%X'92',
;   872          1281  2                                 %X'98',%X'99',%X'9C',%X'9D',%X'9E',%X'A1',
;   873          1282  2                                 %X'B1',%X'B2',%X'FF') ),
;   874          1283  2     Tu_string:          Initial (%ASCID 'TAPE UNIT SELECTED = '),
;   875          1284  2     Serial_number:      word ;
;   876          1285  2
;   877          1286  2
;   878          1287  2
;   879        P 1288  2 STORE_STRINGS ( b1_diag_sts_codes,
;   880        P 1289  2                 'OTHER (WRITE)',
;   881        P 1290  2                 'STATUS (READ)',
;   882        P 1291  2                 'STATUS (READ REVRS)',
;   883        P 1292  2                 'TRANSMISSION (READ)',
;   884          1293  2                 'TRANSMISSION (READ REVRS)' ) ;
;   885          1294  2
;   886        P 1295  2 STORE_STRINGS ( b2_diag_sts_codes,
;   887        P 1296  2                 'DBL TRK CORRECTION (READ)',
;   888        P 1297  2                 'DBL TRK CORRECTION (READ REVRS)',
;   889        P 1298  2                 'SNGL TRK CORRECTION (READ)',
;   890        P 1299  2                 'SNGL TRK CORRECTION (READ REVRS)',
;   891        P 1300  2                 'MEDIA',
;   892        P 1301  2                 'OTHER (READ)',
;   893          1302  2                 'OTHER (READ REVRS)' ) ;
;   894          1303  2
;   895        P 1304  2 STORE_STRINGS ( byte2,
;   896        P 1305  2                 '800 BPI NRZI ENCODING',
;   897        P 1306  2                 '1600 BPI PE ENCODING',
```

```
;  898          1307  2                  '6250 BPI GCR ENCODING' ) ;
;  899          1308  2
;  900        P 1309  2 STORE_STRINGS ( byte17,
;  901        P 1310  2                 'WRITE FAIL F',
;  902        P 1311  2                 'STATISTICS SELECT',
;  903        P 1312  2                 'CLOCK STOPPED',
;  904        P 1313  2                 'BEGINNING OF PREAMBLE',
;  905        P 1314  2                 'DATA READY',
;  906        P 1315  2                 'PREAMBLE ERROR',
;  907        P 1316  2                 'STATUS VALID',
;  908          1317  2                 'VELOC'TY OK' ) ;
;  909          1318  2
;  910          1319  2
;  911        P 1320  2 STORE_STRINGS ( byte18,
;  912        P 132   2                 'FIRST TACH PULSE',
;  913        P 1322  2                 'ECC CONTROLLER PASSED SELF-TEST',
;  914        P 1323  2                 'ECC CONTROLLER FAILED SELF-TEST',
;  915        P 1324  2                 'READ PATH PASSED SELF-TEST',
;  916        P 1325  2                 'READ PATH FAILED SELF-TEST',
;  917        P 1326  2                 'READ CHANNEL TESTS ALL PASSED',
;  918        P 1327  2                 'LAST TACH PULSE',
;  919        P 1328  2                 ''NOT CAPABLE'' FOUND',
;  920        P 1329  2                 ''GCR ID'' FOUND',
;  921        P 1330  2                 ''PE ID'' FOUND',
;  922        P 1331  2                 'BAD STATUS (WRITE TEST)',
;  923        P 1332  2                 'GOOD TAPE MARK FOUND ON TAPE STS',
;  924        P 1333  2                 ''ARA ID'' FOUND',
;  925        P 1334  2                 'TAPE MARK FOUND',
;  926        P 1335  2                 'PREAMBLE END NOT FOUND',
;  927        P 1336  2                 'READ PATH FAULT 1',
;  928        P 1337  2                 'READ PATH FAULT 2',
;  929        P 1338  2                 'UNEXPECTED ''IBG'' IN DATA',
;  930        P 1339  2                 'POSTAMBLE LONG',
;  931        P 1340  2                 'POSTAMBLE SHORT',
;  932          1341  2                 'OK' ) ;
;  933          1342  2
;  934        P 1343  2 STORE_STRINGS ( byte19,
;  935        P 1344  2                 'NOP',
;  936        P 1345  2                 'INTERBLOCK READ',
;  937        P 1346  2                 'TEST PE ID BURST',
;  938        P 1347  2                 'TEST GCR ID BURST',
;  939        P 1348  2                 'TEST ARA ID BURST',
;  940        P 1349  2                 'TEST TAPE MARK',
;  941        P 1350  2                 'TEST ARA BURST',
;  942        P 1351  2                 'NORMAL NON-BOT READ',
;  943        P 1352  2                 'RUN RMC SELF-TEST',
;  944        P 1353  2                 'TEST UNKNOWN ID BURST',
;  945        P 1354  2                 'RUN READ CHANNEL MICROS TEST',
;  946        P 1355  2                 'DIAGNOSTIC READ CMD',
;  947        P 1356  2                 'RUN READ CHANNEL SELF-TEST',
;  948        P 1357  2                 'RUN CLEAR ALL RMC TEST',
;  949        P 1358  2                 'RUN ECC SELF-TEST',
;  950          1359  2                 'FIND GAP' ) ;
;  951          1360  2
;  952        P 1361  2 STORE_STRINGS ( byte25,
;  953        P 1362  2                 'WEAK AMPLITUDE ON PARITY BIT',
;  954        P 1363  2                 'PARITY M8950 NOT DONE',
```

```
;  955        P 1364  2                    'ILLEGAL 5-TO-4 FOR PARITY BIT',
;  956        P 1365  2                    'MARK 2 FOR PARITY BIT',
;  957        P 1366  2                    'END MARK FOR PARITY BIT',
;  958        P 1367  2                    'PE POSTAMBLE FOR PARITY BIT',
;  959        P 1368  2                    'M8950 DATA OUTPUT (PARITY BIT)',
;  960          1369  2                    'ECC CORRECTED OUTPUT (PARITY BIT)' ) ;
;  961          1370  2
;  962        P 1371  2 STORE_STRINGS ( byte30,
;  963        P 1372  2                    'SNGL TRK ECC PERFORMED ON DATA',
;  964        P 1373  2                    '2-TRK ECC PERFORMED ON DATA',
;  965        P 1374  2                    'ECC COULD NOT CORRECT DATA',
;  966        P 1375  2                    'NO POINTER TO TRK IN ERROR',
;  967        P 1376  2                    '"ACRC" DID NOT CHECK',
;  968        P 1377  2                    '"AMTIE" DURING DATA OF RECORD',
;  969        P 1378  2                    'M8951 PROGRAM PARITY ERROR',
;  970          1379  2                    '"CRC" DID NOT CHECK' ) ;
;  971          1380  2
;  972          1381  2
;  973        P 1382  2 STORE_STRINGS ( byte37,
;  974        P 1383  2                    'AMTIE PARITY',
;  975        P 1384  2                    'READ PARITY',
;  976        P 1385  2                    'WCS PARITY',
;  977        P 1386  2                    'TACHOMETER',
;  978        P 1387  2                    'TAPE UNIT PRESENT',
;  979        P 1388  2                    'COMMAND PARITY ERROR',
;  980        P 1389  2                    'WRITE DATA STROBE',
;  981          1390  2                    'STATUS PARITY ERROR' ) ;
;  982          1391  2
;  983        P 1392  2 STORE_STRINGS ( byte39,
;  984        P 1393  2                    'CABLE NOT PRESENT',
;  985        P 1394  2                    'CONTROL PULSE ERROR',
;  986        P 1395  2                    'DATA PULSE ERROR',
;  987        P 1396  2                    'CONTROL PARITY ERROR',
;  988        P 1397  2                    'DATA LATE',
;  989        P 1398  2                    '"CROM" PARITY ERROR',
;  990          1399  2                    'LEVEL 1 PROTOCOL ERROR' ) ;
;  991          1400  2
;  992        P 1401  2 STORE_STRINGS ( byte40,
;  993        P 1402  2                    'XMC WCLK',
;  994        P 1403  2                    'RESIDUAL TO "WMC DR" BUS',
;  995        P 1404  2                    '"ACRC" TO "WMC DR" BUS',
;  996        P 1405  2                    '"CRC" TO "WMC DR" BUS',
;  997        P 1406  2                    'ECC TO "WMC DR" BUS',
;  998        P 1407  2                    'TRANSFER',
;  999        P 1408  2                    '"WMC" NOT DONE',
; 1000          1409  2                    '"XMC" NOT DONE' ) ;
; 1001          1410  2
; 1002        P 1411  2 STORE_STRINGS ( byte41,
; 1003        P 1412  2                    'TU PORT 1 WRITE PATH ENABLE',
; 1004        P 1413  2                    'TU PORT 0 WRITE PATH ENABLE',
; 1005        P 1414  2                    'TU PORT 1 READ PATH ENABLE',
; 1006        P 1415  2                    'TU PORT 0 READ PATH ENABLE',
; 1007        P 1416  2                    'BYTE COUNT TERMINAL COUNT',
; 1008          1417  2                    'SINGLE TAPE UNIT PORT' ) ;
; 1009          1418  2
; 1010        P 1419  2 STORE_STRINGS ( byte42,
; 1011        P 1420  2                    'TU PORT 3 WRITE PATH ENABLE',
```

```
;  1012        P 1421  2                  'TU PORT 2 WRITE PATH ENABLE',
;  1013        P 1422  2                  'TU PORT 3 READ PATH ENABLE',
;  1014          1423  2                  'TU PORT 2 READ PATH ENABLE' ) ;
;  1015          1424  2
;  1016        P 1425  2 STORE_STRINGS ( byte50,
;  1017        P 1426  2                  ''DR'' READ PARITY ERROR',
;  1018        P 1427  2                  ''WMC ROM'' PARITY ERROR',
;  1019        P 1428  2                  'ERROR',
;  1020          1429  2                  ''DR MBD'' PARITY ERROR' ) ;
;  1021          1430  2
;  1022        P 1431  2 STORE_STRINGS ( byte51,
;  1023        P 1432  2                  'TRANSLATOR ''ROM'' PARITY ERROR',
;  1024          1433  2                  ''PE'' WRITE PARITY ERROR' ) ;
;  1025          1434  2
;  1026        P 1435  2 STORE_STRINGS ( byte51_2,
;  1027        P 1436  2                  ''XINI'' SET',
;  1028        P 1437  2                  'WRITE DATA REGISTER PARITY',
;  1029          1438  2                  'POWER OK' ) ;
;  1030          1439  2
;  1031        P 1440  2 STORE_STRINGS ( byte51_3,
;  1032        P 1441  2                  ''USART'' RECEIVER RDY (RX)',
;  1033          1442  2                  'INTERRUPT CLOCK TIME (MCLK)' ) ;
;  1034          1443  2
;  1035        P 1444  2 STORE_STRINGS ( byte52,
;  1036        P 1445  2                  'NO WRITE RING',
;  1037        P 1446  2                  'EOT',
;  1038        P 1447  2                  'BOT',
;  1039        P 1448  2                  'PES',
;  1040        P 1449  2                  'REWINDING',
;  1041        P 1450  2                  'ONLINE',
;  1042        P 1451  2                  'READY ON',
;  1043          1452  2                  'READY' ) ;
;  1044          1453  2
;  1045        P 1454  2 STORE_STRINGS ( byte53,
;  1046        P 1455  2                  'DSE',
;  1047        P 1456  2                  'MOT',
;  1048        P 1457  2                  'LWR',
;  1049        P 1458  2                  'WRITE INHIBIT',
;  1050        P 1459  2                  'WRITE',
;  1051        P 1460  2                  'REVERSE',
;  1052        P 1461  2                  'FORWARD',
;  1053          1462  2                  'MANUAL TEST' ) ;
;  1054          1463  2
;  1055        P 1464  2 STORE_STRINGS ( byte54,
;  1056        P 1465  2                  ''ARA'' ERROR',
;  1057        P 1466  2                  'PEC'
;  1058          1467  2                  'CMD PE' ) ;
;  1059          1468  2
;  1060        P 1469  2 STORE_STRINGS ( byte57,
;  1061        P 1470  2                  'TACH',
;  1062        P 1471  2                  'EOT DET',
;  1063        P 1472  2                  'READ ENABLE',
;  1064        P 1473  2                  'WRITE',
;  1065          1474  2                  'WRITE BIT 4' ) ;
;  1066          1475  2
;  1067          1476  2 Bind
;  1068          1477  2     Drive_sts = emb[82,0,8,0] : VECTOR [,byte],
```

```
;1069          1478  2      Errnum = drive_sts[13] : word,
;1070          1479  2      Unit_number = drive_sts[2] : word,
;1071          1480  2      Gap_count = drive_sts[4] : long,
;1072          1481  2      Byte_count = drive_sts[43] : word,
;1073          1482  2      Pad_count = drive_sts[45] : word,
;1074          1483  2      Frr_code_cnt = drive_sts[47] : word,
;1075          1484  2      Mode_select = .(drive_sts[53])<0,3>,
;1076          1485  2      Speed = .(drive_sts[53])<3,2>,
;1077          1486  2      Sn_b2 = .drive_sts[55],
;1078          1487  2      Sn_b1 = .drive_sts[54] ;
;1079          1488  2
;1080          1489  2  !
;1081          1490  2  ! Output the header.
;1082          1491  2  !
;1083          1492  2  Arglist[0] = CSTRING ('TA78 EXTENDED DRIVE STATUS INFORMATION') ;
;1084          1493  2  Main_hdr = true ;
;1085          1494  2  OUTPUT_LINES (  .fao_strings[0], arglist[0]  ) ;
;1086          1495  2
;1087          1496  2  !
;1088          1497  2  ! Decode and output byte 1,  speed.
;1089          1498  2  !
;1090          1499  2  LABEL_AND_HEX_OUTPUT (1) ;
;1091          1500  2
;1092          1501  2  Arglist[0] = .drive_sts[0] ;
;1093          1502  2  Arglist[1] = %ASCID ' IPS TAPE DRIVE' ;
;1094          1503  2  OUTPUT_LINES (  .fao_strings[11], arglist[0]  ) ;
;1095          1504  2
;1096          1505  2  !
;1097          1506  2  ! Decode and output bytes 2, density.
;1098          1507  2  !
;1099          1508  2  LABEL_AND_HEX_OUTPUT (2) ;
;1100          1509  2
;1101          1510  3  If ( TRANSLATE_BITS (drive_sts[1],bit_msk_11,byte2_desc_tbl,out_arglist,
;1102          1511  3                        fao_string,%REF(0)) )
;1103          1512  2  Then
;1104          1513  2      OUTPUT_LINES (  .fao_string, out_arglist[0]  ) ;
;1105          1514  2
;1106          1515  2
;1107          1516  2  !
;1108          1517  2  ! Decode and output bytes 3/4, unit number lo/hi.
;1109          1518  2  !
;1110          1519  2  LABEL_AND_HEX_OUTPUT (3,4) ;
;1111          1520  2
;1112          1521  2  Arglist[0] = %ASCID 'MSCP UNIT NUMBER = ' ;
;1113          1522  2  Arglist[1] = .unit_number ;
;1114          1523  2
;1115          1524  2  OUTPUT_LINES (  .fao_strings[2], arglist[0]  ) ;
;1116          1525  2
;1117          1526  2  !
;1118          1527  2  ! Decode and output bytes 5/6/7/8, gap count.
;1119          1528  2  !
;1120          1529  2  LABEL_AND_HEX_OUTPUT (5,6,7,8) ;
;1121          1530  2
;1122          1531  2  Arglist[0] = %ASCID 'GAP COUNT = ' ;
;1123          1532  2  Arglist[1] = .gap_count ;
;1124          1533  2
;1125          1534  2  OUTPUT_LINES (  .fao_strings[2], arglist[0]  ) ;
```

```
;  1126          1535  2
;  1127          1536  2 !
;  1128          1537  2 !
;  1129          1538  2 ! Decode and output byte 9, diagnostic mode status byte 1.
;  1130          1539  2 !
;  1131          1540  2 LABEL_AND_HEX_OUTPUT (9) ;
;  1132          1541  2 DIAGNOSTIC_STS_BYTE_DECODE (1,b1_diag_sts_codes,5,
;  1133          1542  2                                 b1_diag_sts_codes_desc_tbl) ;
;  1134          1543  2
;  1135          1544  2 !
;  1136          1545  2 ! Decode and output byte 10, diagnostic mode status byte 2.
;  1137          1546  2 !
;  1138          1547  2 LABEL_AND_HEX_OUTPUT (10) ;
;  1139          1548  2 DIAGNOSTIC_STS_BYTE_DECODE (2,b2_diag_sts_codes,7,
;  1140          1549  2                                 b2_diag_sts_codes_desc_tbl) ;
;  1141          1550  2
;  1142          1551  2 !
;  1143          1552  2 ! If either of the diagnostic status bytes contain a non-zero value
;  1144          1553  2 ! then the rest of the extended drive sts information should not be
;  1145          1554  2 ! output.
;  1146          1555  2 !
;  1147          1556  2 If .drive_sts[8] NEQ 0 OR
;  1148          1557  2    .drive_sts[9] NEQ 0
;  1149          1558  2 Then
;  1150          1559  2     Return ;
;  1151          1560  2
;  1152          1561  2 !
;  1153          1562  2 ! Decode and output byte 11/12, diagnostic mode status bytes 3/4 (unused).
;  1154          1563  2 !
;  1155          1564  2 LABEL_AND_HEX_OUTPUT (11,12) ;
;  1156          1565  2
;  1157          1566  2 !
;  1158          1567  2 ! Decode and output byte 13, opcode save.
;  1159          1568  2 !
;  1160          1569  2 LABEL_AND_HEX_OUTPUT (13) ;
;  1161          1570  2
;  1162          1571  2 Arglist[0] = .sti_string ;
;  1163          1572  2 Arglist[1] = .drive_sts[12] ;
;  1164          1573  2
;  1165          1574  2 OUTPUT_LINES (  .fao_strings[7], arglist[0]  ) ;
;  1166          1575  2
;  1167          1576  2
;  1168          1577  2 !
;  1169          1578  2 ! Decode and output byte 14/15, error number bytes.
;  1170          1579  2 !
;  1171          1580  2 LABEL_AND_HEX_OUTPUT (14,15) ;
;  1172          1581  2 ERROR_NUMBER_BYTE_DECODE (.errnum) ;
;  1173          1582  2
;  1174          1583  2 !
;  1175          1584  2 ! Decode and output byte 16, read micro controller write fail bits.
;  1176          1585  2 !
;  1177          1586  2 LABEL_AND_HEX_OUTPUT (16) ;
;  1178          1587  2 Arglist[0] = %ASCID 'RMC WRITE FAIL BITS' ;
;  1179          1588  2 OUTPUT_LINES (  .fao_strings[3], arglist[0]  ) ;
;  1180          1589  2
;  1181          1590  2 !
;  1182          1591  2 ! Decode and output byte 17, read path status.
```

```
; 1183          1592  2 !
; 1184          1593  2 LABEL_AND_HEX_OUTPUT (17) ;
; 1185          1594  2
; 1186          1595  3 If ( TRANSLATE_BITS (drive_sts[16],bit_msk_0,byte17_desc_tbl,
; 1187          1596  3                         out_arglist,fao_string,%REF(0)) )
; 1188          1597  2 Then
; 1189          1598  2     OUTPUT_LINES (  .fao_string, out_arglist[0]  ) ;
; 1190          1599  2
; 1191          1600  2
; 1192          1601  2 !
; 1193          1602  2 ! Decode and output byte 18, RMC status byte.
; 1194          1603  2 !
; 1195          1604  2 LABEL_AND_HEX_OUTPUT (18) ;
; 1196          1605  2
; 1197          1606  2 Arglist[0] = %ASCID 'UNKNOWN MICROCONTROLLER STS CODE' ;
; 1198          1607  2
; 1199          1608  2 Incr I from 0 to 20 do
; 1200          1609  3     Begin
; 1201          1610  3     If .drive_sts[17] EQL .rmc_sts_codes[.I]
; 1202          1611  3     Then
; 1203          1612  4         Begin
; 1204          1613  4         Arglist[0] = byte18_desc_tbl[.I,0,0,0,0] ;
; 1205          1614  4         Exitloop ;
; 1206          1615  3         End ;
; 1207          1616  2     End ;
; 1208          1617  2
; 1209          1618  2 OUTPUT_LINES (  .fao_strings[3], arglist[0]  ) ;
; 1210          1619  2
; 1211          1620  2
; 1212          1621  2 !
; 1213          1622  2 ! Decode and output byte 19, RMC command.
; 1214          1623  2 !
; 1215          1624  2 LABEL_AND_HEX_OUTPUT (19) ;
; 1216          1625  2
; 1217          1626  2 Arglist[0] = byte19_desc_tbl[.drive_sts[18],0,0,0,0] ;
; 1218          1627  2
; 1219          1628  2 OUTPUT_LINES (  .fao_strings[10], arglist[0]  ) ;
; 1220          1629  2
; 1221          1630  2
; 1222          1631  2 !
; 1223          1632  2 ! Decode and output byte 20, read channel AMTIE sts.
; 1224          1633  2 !
; 1225          1634  2 LABEL_AND_HEX_OUTPUT (20) ;
; 1226          1635  2
; 1227          1636  2 Fao_string = %ASCID '!39< !>AMTIE CHANNEL #!AS' ;
; 1228          1637  2 Incr I from 0 to 7 do
; 1229          1638  3     Begin
; 1230          1639  3     If .(drive_sts[19])<.I>
; 1231          1640  3     Then
; 1232          1641  4         Begin
; 1233          1642  4         Arglist[0] = byte31_desc_tbl[.I,0,0,0,0] ;
; 1234          1643  4         OUTPUT_LINES (  .fao_string, arglist[0]  ) ;
; 1235          1644  3         End ;
; 1236          1645  2     End ;
; 1237          1646  2
; 1238          1647  2
; 1239          1648  2
```

```
;  1240          1649  2 !
;  1241          1650  2 ! Decode and output byte 21, read channel done sts.
;  1242          1651  2 !
;  1243          1652  2 LABEL_AND_HEX_OUTPUT (21) ;
;  1244          1653  2
;  1245          1654  2 Fao_string = %ASCID '!39< !>M895J FAILURE ON CHANNEL #!AS' ;
;  1246          1655  2 Incr I from 0 to 7 do
;  1247          1656  3     Begin
;  1248          1657  3     If .(drive_sts[20])<.I>
;  1249          1658  3     Then
;  1250          1659  4         Begin
;  1251          1660  4         Arglist[0] = byte31_desc_tbl[.I,0,0,0,0] ;
;  1252          1661  4         OUTPUT_LINES (  .fao_string, arglist[0]  ) ;
;  1253          1662  3         End ;
;  1254          1663  2     End ;
;  1255          1664  2
;  1256          1665  2 !
;  1257          1666  2 ! Decode and output byte 22, read channel illegal sts.
;  1258          1667  2 !
;  1259          1668  2 LABEL_AND_HEX_OUTPUT (22) ;
;  1260          1669  2
;  1261          1670  2 Arglist[0] = %ASCID 'READ CHANNEL ILLEGAL STS (CH 7:0)' ;
;  1262          1671  2 OUTPUT_LINES (  .fao_strings[3], arglist[0]  ) ;
;  1263          1672  2
;  1264          1673  2 !
;  1265          1674  2 ! Decode and output byte 23, read channel mark 2 sts.
;  1266          1675  2 !
;  1267          1676  2 LABEL_AND_HEX_OUTPUT (23) ;
;  1268          1677  2
;  1269          1678  2 Fao_string = %ASCID 'MARK 2 ERROR, CHANNEL !ZB (M8950)' ;
;  1270          1679  2 Incr I from 0 to 7 do
;  1271          1680  3     Begin
;  1272          1681  3     If .(drive_sts[22])<.I>
;  1273          1682  3     Then
;  1274          1683  4         Begin
;  1275          1684  4         Arglist[0] = .(drive_sts[22])<.I> ;
;  1276          1685  4         OUTPUT_LINES (  .fao_string, arglist[0]  ) ;
;  1277          1686  3         End ;
;  1278          1687  2     End ;
;  1279          1688  2
;  1280          1689  2 !
;  1281          1690  2 ! Decode and output byte 24, read channel end sts.
;  1282          1691  2 !
;  1283          1692  2 LABEL_AND_HEX_OUTPUT (24) ;
;  1284          1693  2
;  1285          1694  2 Arglist[0] = %ASCID 'END MARK FOR READ CHANNELS 7:0' ;
;  1286          1695  2 OUTPUT_LINES (  .fao_strings[3], arglist[0]  ) ;
;  1287          1696  2
;  1288          1697  2 !
;  1289          1698  2 ! Decode and output byte 25, read channel parity sts.
;  1290          1699  2 !
;  1291          1700  2 LABEL_AND_HEX_OUTPUT (25) ;
;  1292          1701  2
;  1293          1702  3 If ( TRANSLATE_BITS (drive_sts[24],bit_msk_0,byte25_desc_tbl,
;  1294          1703  3                       out_arglist,fao_string,%REF(0)) )
;  1295          1704  2 Then
;  1296          1705  2     OUTPUT_LINES (  .fao_string, out_arglist[0]  ) ;
```

```
;  1297          1706  2
;  1298          1707  2 !
;  1299          1708  2 ! Decode and output byte 26, read channel pe postamble detect.
;  1300          1709  2 !
;  1301          1710  2 LABEL_AND_HEX_OUTPUT (26) ;
;  1302          1711  2
;  1303          1712  2 Arglist[0] = %ASCID 'READ CHANNEL PE POSTAMBLE DETECT' ;
;  1304          1713  2 OUTPUT_LINES (  .fao_strings[3], arglist[0]  ) ;
;  1305          1714  2
;  1306          1715  2
;  1307          1716  2 !
;  1308          1717  2 ! Decode and output byte 27, read channel data.
;  1309          1718  2 !
;  1310          1719  2 LABEL_AND_HEX_OUTPUT (27) ;
;  1311          1720  2
;  1312          1721  2 Arglist[0] = %ASCID 'DATA FROM READ CHANNELS TO ECC' ;
;  1313          1722  2 OUTPUT_LINES (  .fao_strings[3], arglist[0]  ) ;
;  1314          1723  2
;  1315          1724  2 !
;  1316          1725  2 ! Decode and output byte 28, CRC byte.
;  1317          1726  2 !
;  1318          1727  2 LABEL_AND_HEX_OUTPUT (28) ;
;  1319          1728  2
;  1320          1729  2 Arglist[0] = %ASCID 'CRC CHECKER OUTPUT BITS' ;
;  1321          1730  2 OUTPUT_LINES (  .fao_strings[3], arglist[0]  ) ;
;  1322          1731  2
;  1323          1732  2 !
;  1324          1733  2 ! Decode and output byte 29, corrected data.
;  1325          1734  2 !
;  1326          1735  2 LABEL_AND_HEX_OUTPUT (29) ;
;  1327          1736  2
;  1328          1737  2 Arglist[0] = %ASCID 'CORRECTED DATA (ECC TO CRC)' ;
;  1329          1738  2 OUTPUT_LINES (  .fao_strings[3], arglist[0]  ) ;
;  1330          1739  2
;  1331          1740  2 !
;  1332          1741  2 ! Decode and output byte 30, ECC sts byte.
;  1333          1742  2 !
;  1334          1743  2 LABEL_AND_HEX_OUTPUT (30) ;
;  1335          1744  2
;  1336          1745  3 If ( TRANSLATE_BITS (drive_sts[29],bit_msk_0,byte30_desc_tbl,
;  1337          1746  3                       out_arglist,fao_string,%REF(0)) )
;  1338          1747  2 Then
;  1339          1748  2     OUTPUT_LINES (  .fao_string, out_arglist[0]  ) ;
;  1340          1749  2
;  1341          1750  2
;  1342          1751  2 !
;  1343          1752  2 ! Decode and output byte 31, read channels 0 to 1 TIE.
;  1344          1753  2 !
;  1345          1754  2 LABEL_AND_HEX_OUTPUT (31) ;
;  1346          1755  2 READ_CHANNEL_TIE_DECODE (.drive_sts[30],%REF(0)) ;
;  1347          1756  2
;  1348          1757  2 !
;  1349          1758  2 ! Decode and output byte 32, read channels 2 and 3 TIE.
;  1350          1759  2 !
;  1351          1760  2 LABEL_AND_HEX_OUTPUT (32) ;
;  1352          1761  2 READ_CHANNEL_TIE_DECODE (.drive_sts[31],%REF(1)) ;
;  1353          1762  2
```

```
;  1354          1763  2 !
;  1355          1764  2 ! Decode and output byte 33, read 4 and 5 TIE.
;  1356          1765  2 !
;  1357          1766  2 LABEL_AND_HEX_OUTPUT (33) ;
;  1358          1767  2 READ_CHANNEL_TIE_DECODE (.drive_sts[32],%REF(2)) ;
;  1359          1768  2
;  1360          1769  2 !
;  1361          1770  2 ! Decode and output byte 34, read 6 and 7 TIE.
;  1362          1771  2 !
;  1363          1772  2 LABEL_AND_HEX_OUTPUT (34) ;
;  1364          1773  2 READ_CHANNEL_TIE_DECODE (.drive_sts[33],%REF(3)) ;
;  1365          1774  2
;  1366          1775  2 !
;  1367          1776  2 ! Decode and output byte 35, read channel p TIE and TIE bus.
;  1368          1777  2 !
;  1369          1778  2 LABEL_AND_HEX_OUTPUT (35) ;
;  1370          1779  2
;  1371          1780  2 Fao_string = %ASCID '!39< !>CHP TIE BUS !AS' ;
;  1372          1781  2 Incr I from 0 to 3 do
;  1373          1782  3     Begin
;  1374          1783  3     If .(drive_sts[34])<.I>
;  1375          1784  3     Then
;  1376          1785  4         Begin
;  1377          1786  4         Arglist[0] = byte31_desc_tbl[.I,0,0,0,0] ;
;  1378          1787  4         OUTPUT_LINES (  .fao_string, arglist[0]  ) ;
;  1379          1788  3         End ;
;  1380          1789  2     End ;
;  1381          1790  2
;  1382          1791  2 Fao_string = %ASCID '!39< !>TIE BUS = !XB(X)' ;
;  1383          1792  2 Arglist[0] = .(drive_sts[34])<7,4> ;
;  1384          1793  2
;  1385          1794  2 OUTPUT_LINES (  .fao_string, arglist[0]  ) ;
;  1386          1795  2
;  1387          1796  2 !
;  1388          1797  2 ! Decode and output byte 36, AMTIE byte.
;  1389          1798  2 !
;  1390          1799  2 LABEL_AND_HEX_OUTPUT (36) ;
;  1391          1800  2
;  1392          1801  2 Arglist[0] = %ASCID 'TAPE UNIT BUS LINE AMTIE 7:0' ;
;  1393          1802  2 OUTPUT_LINES (  .fao_strings[3], arglist[0]  ) ;
;  1394          1803  2
;  1395          1804  2 !
;  1396          1805  2 ! Decode and output byte 37, tape unit port sts.
;  1397          1806  2 !
;  1398          1807  2 LABEL_AND_HEX_OUTPUT (37) ;
;  1399          1808  2
;  1400          1809  3 If ( TRANSLATE_BITS (drive_sts[36],bit_msk_0,byte37_desc_tbl,
;  1401          1810  3                       out_arglist,fao_string,%REF(0)) )
;  1402          1811  2 Then
;  1403          1812  2     OUTPUT_LINES (  .fao_string, out_arglist[0]  ) ;
;  1404          1813  2
;  1405          1814  2
;  1406          1815  2 !
;  1407          1816  2 ! Decode and output byte 38, tu port read data.
;  1408          1817  2 !
;  1409          1818  2 LABEL_AND_HEX_OUTPUT (38) .
;  1410          1819  2
```

```
;  1411          1820  2 Arglist[0] = %ASCID 'TU BUS LINE READ DATA 7:0' ;
;  1412          1821  2 OUTPUT_LINES (  .fao_strings[3], arglist[0]  ) ;
;  1413          1822  2
;  1414          1823  2 !
;  1415          1824  2 ! Decode and output byte 39, STI bus errors.
;  1416          1825  2 !
;  1417          1826  2 LABEL_AND_HEX_OUTPUT (39) ;
;  1418          1827  2
;  1419          1828  3 If ( TRANSLATE_BITS (drive_sts[38],bit_msk_8,byte39_desc_tbl,
;  1420          1829  3                       out_arglist,fao_string,%REF(0)) )
;  1421          1830  2 Then
;  1422          1831  2     OUTPUT_LINES ( .fao_string, out_arglist[0]  ) ;
;  1423          1832  2
;  1424          1833  2 !
;  1425          1834  2 ! Decode and output byte 40, write microcontroller sts.
;  1426          1835  2 !
;  1427          1836  2 LABEL_AND_HEX_OUTPUT (40) ;
;  1428          1837  2
;  1429          1838  3 If ( TRANSLATE_BITS (drive_sts[39],bit_msk_0,byte40_desc_tbl,
;  1430          1839  3                       out_arglist,fao_string,%REF(0)) )
;  1431          1840  2 Then
;  1432          1841  2     OUTPUT_LINES ( .fao_string, out_arglist[0]  ) ;
;  1433          1842  2
;  1434          1843  2 !
;  1435          1844  2 ! Decode and output byte 41, tape unit port 0/1 select.
;  1436          1845  2 !
;  1437          1846  2 LABEL_AND_HEX_OUTPUT (41) ;
;  1438          1847  2
;  1439          1848  2 Arglist[0] = .tu_string ;
;  1440          1849  2 Arglist[1] = .(drive_sts[40])<0,2> ;
;  1441          1850  2
;  1442          1851  2 OUTPUT_LINES (  .fao_strings[2], arglist[0]  ) ;
;  1443          1852  2
;  1444          1853  3 If ( TRANSLATE_BITS (drive_sts[40],bit_msk_3,byte41_desc_tbl,
;  1445          1854  3                       out_arglist,fao_string,%REF(0)) )
;  1446          1855  2 Then
;  1447          1856  2     OUTPUT_LINES (  .fao_string, out_arglist[0]  ) ;
;  1448          1857  2
;  1449          1858  2 !
;  1450          1859  2 ! Decode and output byte 42, tape unit port 2/3 select.
;  1451          1860  2 !
;  1452          1861  2 LABEL_AND_HEX_OUTPUT (42) ;
;  1453          1862  2
;  1454          1863  2 Arglist[0] = .tu_string ;
;  1455          1864  2 Arglist[1] = .(drive_sts[41])<0,2> ;
;  1456          1865  2
;  1457          1866  2 OUTPUT_LINES (  .fao_strings[2], arglist[0]  ) ;
;  1458          1867  2
;  1459          1868  3 If ( TRANSLATE_BITS (drive_sts[41],bit_msk_10,byte42_desc_tbl,
;  1460          1869  3                       out_arglist,fao_string,%REF(0)) )
;  1461          1870  2 Then
;  1462          1871  2     OUTPUT_LINES (  .fao_string, out_arglist[0]  ) ;
;  1463          1872  2
;  1464          1873  2
;  1465          1874  2 !
;  1466          1875  2 ! Decode and output byte 43, write data.
;  1467          1876  2 !
```

```
; 1468          1877  2 LABEL_AND_HEX_OUTPUT (43) ;
; 1469          1878  2
; 1470          1879  2 Arglist[0] = %ASCID 'R/W DATA, INTERMEDIATE DRD BUS' ;
; 1471          1880  2 OUTPUT_LINES (  .fao_strings[3], arglist[0]  ) ;
; 1472          1881  2
; 1473          1882  2 !
; 1474          1883  2 ! Decode and output byte 44/45, byte count lo/hi.
; 1475          1884  2 !
; 1476          1885  2 LABEL_AND_HEX_OUTPUT (44,45) ;
; 1477          1886  2
; 1478          1887  2 Fao_string = %ASCID '!39< !>BYTE COUNT = !ZW.' ;
; 1479          1888  2 Arglist[0] = .byte_count ;
; 1480          1889  2
; 1481          1890  2 OUTPUT_LINES (  .fao_string, arglist[0]  ) ;
; 1482          1891  2
; 1483          1892  2 !
; 1484          1893  2 ! Decode and output byte 46/47, pad count lo/hi.
; 1485          1894  2 !
; 1486          1895  2 LABEL_AND_HEX_OUTPUT (46,47) ;
; 1487          1896  2
; 1488          1897  2 Fao_string = %ASCID '!39< !>PAD COUNTER = !ZW.' ;
; 1489          1898  2 Arglist[0] = .pad_count ;
; 1490          1899  2
; 1491          1900  2 OUTPUT_LINES (  .fao_string, arglist[0]  ) ;
; 1492          1901  2
; 1493          1902  2 !
; 1494          1903  2 ! Decode and output byte 48/49, error code counter lo/hi.
; 1495          1904  2 !
; 1496          1905  2 LABEL_AND_HEX_OUTPUT (48,49) ;
; 1497          1906  2
; 1498          1907  2 Fao_string = .fao_strings[3] ;
; 1499          1908  2 Case .err_code_cnt from -5 to 0 of
; 1500          1909  2     Set
; 1501          1910  2     [0]: Arglist[0] = %ASCID 'SUCCESSFUL COMPLETION' ;
; 1502          1911  2     [-1]:
; 1503          1912  3         Begin
; 1504          1913  3         Arglist[0] = %ASCID 'READ SKIP COUNT WAS -' ;
; 1505          1914  3         Arglist[1] = %ASCID '>4 IN 0 CORE DUMP OR' ;
; 1506          1915  3         Arglist[2] = %ASCID '>3 IN 10 COMPATIBLE OR ' ;
; 1507          1916  3         Arglist[3] = %ASCID '>8 IN 10 HIGH-DENSITY COMPATIBLE' ;
; 1508          1917  3
; 1509          1918  3         Fao_string = %ASCID '!39< !>!AS!/!39< !>!AS!/!39< !>!AS!/!39< !>!AS' ;
; 1510          1919  3         End ;
; 1511          1920  2     [-2]:
; 1512          1921  3         Begin
; 1513          1922  3         Arglist[0] = %ASCID 'READ SKIP COUNT >1 IN' ;
; 1514          1923  3         Arglist[1] = %ASCID '11 or 15 NORMAL' ;
; 1515          1924  3
; 1516          1925  3         Fao_string = %ASCID '!39< !>!AS!/!39< !>!AS' ;
; 1517          1926  3         End ;
; 1518          1927  2
; 1519          1928  2     [-3]: Arglist[0] = %ASCID 'FORMAT CODE >6' ;
; 1520          1929  2
; 1521          1930  2     [-4]: Arglist[0] = %ASCID ''WMC'' SELF-TEST DIAGNOSTIC ERROR' ;
; 1522          1931  2
; 1523          1932  2     [-5]: Arglist[0] = %ASCID 'READ OVERRUN OR WRITE FAULT' ;
; 1524          1933  2
```

```
;  1525          1934  2      [OUTRANGE]: Arglist[0] = %ASCID 'UNKNOWN ERROR CODE' ;
;  1526          1935  2      Tes ;
;  1527          1936  2
;  1528          1937  2  OUTPUT_LINES (  .fao_string, arglist[0]  ) ;
;  1529          1938  2
;  1530          1939  2  !
;  1531          1940  2  ! Decode and output byte 50, write microcontroller errors.
;  1532          1941  2  !
;  1533          1942  2  LABEL_AND_HEX_OUTPUT (50) ;
;  1534          1943  2
;  1535          1944  3  If ( TRANSLATE_BITS (drive_sts[49],bit_msk_6,byte50_desc_tbl,
;  1536          1945  3                        out_arglist,fao_string,%REF(0)) )
;  1537          1946  2  Then
;  1538          1947  2      OUTPUT_LINES (  .fao_string, out_arglist[0]  ) ;
;  1539          1948  2
;  1540          1949  2  !
;  1541          1950  2  ! Decode and output byte 51, interrupt sts.
;  1542          1951  2  !
;  1543          1952  2  LABEL_AND_HEX_OUTPUT (51) ;
;  1544          1953  2
;  1545          1954  3  If ( TRANSLATE_BITS (drive_sts[50],bit_msk_12,byte51_desc_tbl,
;  1546          1955  3                        out_arglist,fao_string,%REF(0)) )
;  1547          1956  2  Then
;  1548          1957  2      OUTPUT_LINES (  .fao_string, out_arglist[0]  ) ;
;  1549          1958  2
;  1550          1959  2  If NOT .(drive_sts[50])<2>
;  1551          1960  2  Then
;  1552          1961  3      Begin
;  1553          1962  3      Arglist[0] = %ASCID ''USART'' TRANSMITTER RDY (TX)' ;
;  1554          1963  3      OUTPUT_LINES (  .fao_strings[3], arglist[0]  ) ;
;  1555          1964  2      End ;
;  1556          1965  2
;  1557          1966  3  If ( TRANSLATE_BITS (drive_sts[50],bit_msk_7,byte51_2_desc_tbl,
;  1558          1967  3                        out_arglist,fao_string,%REF(0)) )
;  1559          1968  2  Then
;  1560          1969  2      OUTPUT_LINES (  .fao_string, out_arglist[0]  ) ;
;  1561          1970  2
;  1562          1971  3  If ( TRANSLATE_BITS (drive_sts[50],bit_msk_3,byte51_3_desc_tbl,
;  1563          1972  3                        out_arglist,fao_string,%REF(1)) )
;  1564          1973  2  Then
;  1565          1974  2      OUTPUT_LINES (  .fao_string, out_arglist[0]  ) ;
;  1566          1975  2
;  1567          1976  2
;  1568          1977  2  !
;  1569          1978  2  ! Decode and output byte 52, TU78 sts.
;  1570          1979  2  !
;  1571          1980  2  LABEL_AND_HEX_OUTPUT (52) ;
;  1572          1981  2
;  1573          1982  3  If ( TRANSLATE_BITS (drive_sts[51],bit_msk_0,byte52_desc_tbl,
;  1574          1983  3                        out_arglist,fao_string,%REF(0)) )
;  1575          1984  2  Then
;  1576          1985  2      OUTPUT_LINES (  .fao_string, out_arglist[0]  ) ;
;  1577          1986  2
;  1578          1987  2  !
;  1579          1988  2  ! Decode and output byte 53, MIS sts A.
;  1580          1989  2  !
;  1581          1990  2  LABEL_AND_HEX_OUTPUT (53) ;
```

```
;  1582          1991  2
;  1583          1992  3 If ( TRANSLATE_BITS (drive_sts[52],bit_msk_0,byte53_desc_tbl,
;  1584          1993  3                          out_arglist,fao_string,%REF(0)) )
;  1585          1994  2 Then
;  1586          1995  2     OUTPUT_LINES (  .fao_string, out_arglist[0]  ) ;
;  1587          1996  2
;  1588          1997  2
;  1589          1998  2 !
;  1590          1999  2 ! Decode and output byte 54, MIA sts B.
;  1591          2000  2 !
;  1592          2001  2 LABEL_AND_HEX_OUTPUT (54) ;
;  1593          2002  2
;  1594          2003  2 Case mode_select from 3 to 7 of
;  1595          2004  2     Set
;  1596          2005  2     [3]: Arglist[0] = %ASCID 'POSITION 2 (MAINT)' ;
;  1597          2006  2     [5]: Arglist[0] = %ASCID 'POSITION 1 (MAINT)' ;
;  1598          2007  2     [6]: Arglist[0] = %ASCID 'POSITION 0 (NORMAL)' ;
;  1599          2008  2     [7]: Arglist[0] = %ASCID 'POSITION 3 (MAINT)' ;
;  1600          2009  2
;  1601          2010  2     [INRANGE,OUTRANGE]: Arglist[1] = %ASCID 'UNKNOWN MODE SELECTION' ;
;  1602          2011  2     Tes ;
;  1603          2012  2
;  1604          2013  2 Arglist[1] = %ASCID '125 IPS TAPE DRIVE' ;
;  1605          2014  2
;  1606          2015  2 OUTPUT_LINES (.fao_strings[3], arglist[0],
;  1607          2016  2                .fao_strings[3], arglist[1]  ) ;
;  1608          2017  2
;  1609          2018  2
;  1610          2019  3 If ( TRANSLATE_BITS (drive_sts[53],bit_msk_9,byte54_desc_tbl,
;  1611          2020  3                          out_arglist,fao_string,%REF(0)) )
;  1612          2021  2 Then
;  1613          2022  2     OUTPUT_LINES (  .fao_string, out_arglist[0]  ) ;
;  1614          2023  2
;  1615          2024  2
;  1616          2025  2 !
;  1617          2026  2 ! Decode and output byte 55/5c, serial number A/B.
;  1618          2027  2 !
;  1619          2028  2 LABEL_AND_HEX_OUTPUT (55,56) ;
;  1620          2029  2
;  1621          2030  2 Arglist[0] = %ASCID 'TAPE UNIT SERIAL #' ;
;  1622          2031  2
;  1623          2032  2 Serial_number<0,8> = sn_b2 ;
;  1624          2033  2 Serial_number<8,8> = sn_b1 ;
;  1625          2034  2 Arglist[1] = CONVERT_BCD_NUMBER (.serial_number, 4) ;
;  1626          2035  2
;  1627          2036  2 OUTPUT_LINES (  .fao_strings[12], arglist[0]  ) ;
;  1628          2037  2
;  1629          2038  2 !
;  1630          2039  2 ! Decode and output byte 57, tape unit diagnostics.
;  1631          2040  2 !
;  1632          2041  2 LABEL_AND_HEX_OUTPUT (57) ;
;  1633          2042  2
;  1634          2043  2 Arglist[0] = %ASCID 'AMTIE THRESHOLD FIELD = ' ;
;  1635          2044  2 Arglist[1] = .(drive_sts[56])<0,2> ;
;  1636          2045  2
;  1637          2046  2 OUTPUT_LINES (  .fao_strings[2], arglist[0]  ) ;
;  1638          2047  2
```

```
: 1639          2048  3 If (TRANSLATE_BITS (drive_sts[56],bit_msk_2,byte57_desc_tbl,
: 1640          2049  3                      out_arglist,fao_string,%REF(0)))
: 1641          2050  2 Then
: 1642          2051  2     OUTPUT_LINES (  .fao_string. out_arglist[0]  ) ;
: 1643          2052  2
: 1644          2053  2 !
: 1645          2054  2 ! Output the spare bytes (58/59) headers.
: 1646          2055  2 !
: 1647          2056  2 LABEL_AND_HEX_OUTPUT (58,59) ;
: 1648          2057  2
: 1649          2058  2 Return ;
: 1650          2059  1 End ;   ! Routine


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

43 45 4C 45 53 20 54 49 4E 55 20 45 50 41 54  00D58 P.AGY:   .ASCII  \TAPE UNIT SELECTED = \<0><0><0>
                  00 00 00 20 3D 20 44 45 54  00D67
                                 010E0015     00D70 P.AGX:   .LONG   17694741
                                 00000000'    00D74          .ADDRESS P.AGY
00 00 29 45 54 49 52 57 28 20 52 45 48 54 4F  00D78 P.AGZ:   .ASCII  \OTHER (WRITE)\<0><0><0>
                                             00  00D87
00 00 29 44 41 45 52 28 20 53 55 54 41 54 53  00D88 P.AHA:   .ASCII  \STATUS (READ)\<0><0><0>
                                             00  00D97
45 52 20 44 41 45 52 28 20 53 55 54 41 54 53  00D98 P.AHB:   .ASCII  \STATUS (READ REVRS)\<0>
                                    00 29 53 52 56  00DA7
52 28 20 4E 4F 49 53 53 49 4D 53 4E 41 52 54  00DAC P.AHC:   .ASCII  \TRANSMISSION (READ)\<0>
                                    00 29 44 41 45  00DBB
52 28 20 4E 4F 49 53 53 49 4D 53 4E 41 52 54  00DC0 P.AHD:   .ASCII  \TRANSMISSION (READ REVRS)\<0><0><0>
   00 00 00 29 53 52 56 45 52 20 44 41 45     00DCF
54 43 45 52 52 4F 43 20 4B 52 54 20 4C 42 44  00DDC P.AHE:   .ASCII  \DBL TRK CORRECTION (READ)\<0><0><0>
   00 00 00 29 44 41 45 52 28 20 4E 4F 49     00DEB
54 43 45 52 52 4F 43 20 4B 52 54 20 4C 42 44  00DF8 P.AHF:   .ASCII  \DBL TRK CORRECTION (READ REVRS)\<0>
53 52 56 45 52 20 44 41 45 52 28 20 4E 4F 49  00E07
                                 00 29 53 52 56  00E16
43 45 52 52 4F 43 20 4B 52 54 20 4C 47 4E 53  00E18 P.AHG:   .ASCII  \SNGL TRK CORRECTION (READ)\<0><0>
   00 00 29 44 41 45 52 28 20 4E 4F 49 54     00E27
43 45 52 52 4F 43 20 4B 52 54 20 4C 47 4E 53  00E34 P.AHH:   .ASCII  \SNGL TRK CORRECTION (READ REVRS)\
52 56 45 52 20 44 41 45 52 28 20 4E 4F 49 54  00E43
                                 29 53 52 56  00E52
                  00 00 00 41 49 44 45 4D  00E54 P.AHI:   .ASCII  \MEDIA\<0><0><0>
      29 44 41 45 52 28 20 52 45 48 54 4F  00E5C P.AHJ:   .ASCII  \OTHER (READ)\
56 45 52 20 44 41 45 52 28 20 52 45 48 54 4F  00E68 P.AHK:   .ASCII  \OTHER (READ REVRS)\<0><0>
                                 00 00 29 53 52  00E77
4E 45 20 49 5A 52 4E 20 49 50 42 20 30 30 38  00E7C P.AHL:   .ASCII  \800 BPI NRZI ENCODING\<0><0><0>
                  00 00 00 47 4E 49 44 4F 43  00E88
43 4E 45 20 45 50 20 49 50 42 20 30 30 36 31  00E94 P.AHM:   .ASCII  \1600 BPI PE ENCODING\
                                 47 4E 49 44 4F  00EA3
4E 45 20 52 43 47 20 49 50 42 20 30 35 32 36  00EA8 P.AHN:   .ASCII  \6250 BPI GCR ENCODING\<0><0><0>
                  00 00 00 47 4E 49 44 4F 43  00EB7
      50 20 4C 49 41 46 20 45 54 49 52 57  00EC0 P.AHO:   .ASCII  \WRITE FAIL P\
45 4C 45 53 20 53 43 49 54 53 49 54 41 54 53  00ECC P.AHP:   .ASCII  \STATISTICS SELECT\<0><0><0>
                                 00 00 00 54 43  00EDB
00 00 44 45 50 50 4F 54 53 20 4B 43 4F 4C 43  00EE0 P.AHQ:   .ASCII  \CLOCK STOPPED\<0><0><0>
                                             00  00EEF
52 50 20 46 4F 20 47 4E 49 4E 4E 49 47 45 42  00EF0 P.AHR:   .ASCII  \BEGINNING OF PREAMBLE\<0><0><0>
```

```
                              00 00 00 45 4C 42 4D 41 45 00EFF
                  00 00 59 44 41 45 52 20 41 54 41 44 00F08 P.AHS:  .ASCII  \DATA READY\<0><0>
00 52 4F 52 52 45 20 45 4C 42 4D 41 45 52 50 00F14 P.AHT:  .ASCII  \PREAMBLE ERROR\<0><0>
                                               00 00F23
         44 49 4C 41 56 20 53 55 54 41 54 53 00F24 P.AHU:  .ASCII  \STATUS VALID\
         00 4B 4F 20 59 54 49 43 4F 4C 45 56 00F30 P.AHV:  .ASCII  \VELOCITY OK\<0>
53 4C 55 50 20 48 43 41 54 20 54 53 52 49 46 00F3C P.AHW:  .ASCII  \FIRST TACH PULSE\
                                               45 00F4B
20 52 45 4C 4C 4F 52 54 4E 4F 43 20 43 43 45 00F4C P.AHX:  .ASCII  \ECC CONTROLLER PASSED SELF-TEST\<0>
53 45 54 2D 46 4C 45 53 20 44 45 53 53 41 50 00F5B
                                            00 54 00F6A
20 52 45 4C 4C 4F 52 54 4E 4F 43 20 43 43 45 00F6C P.AHY:  .ASCII  \ECC CONTROLLER FAILED SELF-TEST\<0>
53 45 54 2D 46 4C 45 53 20 44 45 4C 49 41 46 00F7B
                                            00 54 00F8A
45 53 53 41 50 20 48 54 41 50 20 44 41 45 52 00F8C P.AHZ:  .ASCII  \READ PATH PASSED SELF-TEST\<0><0>
      00 00 54 53 45 54 2D 46 4C 45 53 20 44 00F9B
45 4C 49 41 46 20 48 54 41 50 20 44 41 45 52 00FA8 P.AIA:  .ASCII  \READ PATH FAILED SELF-TEST\<0><0>
      00 00 54 53 45 54 2D 46 4C 45 53 20 44 00FB7
45 54 20 4C 45 4E 4E 41 48 43 20 44 41 45 52 00FC4 P.AIB:  .ASCII  \READ CHANNEL TESTS ALL PASSED\<0><0>
00 44 45 53 53 41 50 20 4C 4C 41 20 53 54 53 00FD3
                                               00 00FE2
                                               00 00FE3         .ASCII  <0>
45 53 4C 55 50 20 48 43 41 54 20 54 53 41 4C 00FE4 P.AIC:  .ASCII  \LAST TACH PULSE\<0>
                                               00 00FF3
46 20 22 45 4C 42 41 50 41 43 20 54 4F 4E 22 00FF4 P.AID:  .ASCII  \"NOT CAPABLE" FOUND\<0>
                              00 44 4E 55 4F 01003
00 44 4E 55 4F 46 20 22 44 49 20 52 43 47 22 01008 P.AIE:  .ASCII  \"GCR ID" FOUND\<0><0>
                                               00 01017
00 00 44 4E 55 4F 46 20 22 44 49 20 45 50 22 01018 P.AIF:  .ASCII  \"PE ID" FOUND\<0><0><0>
                                               00 01027
49 52 57 28 20 53 55 54 41 54 53 20 44 41 42 01028 P.AIG:  .ASCII  \BAD STATUS (WRITE TEST)\<0>
                  00 29 54 53 45 54 20 45 54 01037
20 4B 52 41 4D 20 45 50 41 54 20 44 4F 4F 47 01040 P.AIH:  .ASCII  \GOOD TAPE MARK FOUND ON TAPE STS\
53 20 45 50 41 54 20 4E 4F 20 44 4E 55 4F 46 0104F
                                            53 54 0105E
00 44 4E 55 4F 46 20 22 44 49 20 41 52 41 22 01060 P.AII:  .ASCII  \"ARA ID" FOUND\<0><0>
                                               00 0106F
44 4E 55 4F 46 20 4B 52 41 4D 20 45 50 41 54 01070 P.AIJ:  .ASCII  \TAPE MARK FOUND\<0>
                                               00 0107F
4F 4E 20 44 4E 45 20 45 4C 42 4D 41 45 52 50 01080 P.AIK:  .ASCII  \PREAMBLE END NOT FOUND\<0><0>
                  00 00 44 4E 55 4F 46 20 54 0108F
54 4C 55 41 46 20 48 54 41 50 20 44 41 45 52 01098 P.AIL:  .ASCII  \READ PATH FAULT 1\<0><0><0>
                              00 00 00 31 20 010A7
54 4C 55 41 46 20 48 54 41 50 20 44 41 45 52 010AC P.AIM:  .ASCII  \READ PATH FAULT 2\<0><0><0>
                              00 00 00 32 20 010BB
47 42 49 22 20 44 45 54 43 45 50 58 45 4E 55 010C0 P.AIN:  .ASCII  \UNEXPECTED "IBG" IN DATA\
                  41 54 41 44 20 4E 49 20 22 010CF
00 47 4E 4F 4C 20 45 4C 42 4D 41 54 53 4F 50 010D0 P.AIO:  .ASCII  \POSTAMBLE LONG\<0><0>
                                               00 010E7
54 52 4F 48 53 20 45 4C 42 4D 41 54 53 4F 50 010E8 P.AIP:  .ASCII  \POSTAMBLE SHORT\<0>
                                               00 010F7
                                    00 00 4B 4F 010F8 P.AIQ:  .ASCII  \OK\<0><0>
                                    00 50 4F 4E 010FC P.AIR:  .ASCII  \NOP\<0>
44 41 45 52 20 4B 43 4F 4C 42 52 45 54 4E 49 01100 P.AIS:  .ASCII  \INTERBLOCK READ\<0>
                                               00 0110F
53 52 55 42 20 44 49 20 45 50 20 54 53 45 54 01110 P.AIT:  .ASCII  \TEST PE ID BURST\
                                               54 0111F
```

```
52 55 42 20 44 49 20 52 43 47 20 54 53 45 54 01120 P.AIU:  .ASCII  \TEST GCR ID BURST\<0><0><0>
                              00 00 00 54 53 0112F
52 55 42 20 44 49 20 41 52 41 20 54 53 45 54 01134 P.AIV:  .ASCII  \TEST ARA ID BURST\<0><0><0>
                              00 00 00 54 53 01143
00 4B 52 41 4D 20 45 50 41 54 20 54 53 45 54 01148 P.AIW:  .ASCII  \TEST TAPE MARK\<0><0>
                                                 00 01157
00 54 53 52 55 42 20 41 52 41 20 54 53 45 54 01158 P.AIX:  .ASCII  \TEST ARA BURST\<0><0>
                                                 00 01167
20 54 4F 42 2D 4E 4F 4E 20 4C 41 4D 52 4F 4E 01168 P.AIY:  .ASCII  \NORMAL NON-BOT READ\<0>
                              00 44 41 45 52 01177
45 54 2D 46 4C 45 53 20 43 4D 52 20 4E 55 52 0117C P.AIZ:  .ASCII  \RUN RMC SELF-TEST\<0><0><0>
                              00 00 00 54 53 01188
44 49 20 4E 57 4F 4E 4B 4E 55 20 54 53 45 54 01190 P.AJA:  .ASCII  \TEST UNKNOWN ID BURST\<0><0><0>
                        00 00 00 54 53 52 55 42 20 0119F
45 4E 4E 41 48 43 20 44 41 45 52 20 4E 55 52 011A8 P.AJB:  .ASCII  \RUN READ CHANNEL MICROS TEST\
      54 53 45 54 20 53 4F 52 43 49 4D 20 4C 011B7
44 41 45 52 20 43 49 54 53 4F 4E 47 41 49 44 011C4 P.AJC:  .ASCII  \DIAGNOSTIC READ CMD\<0>
                              00 44 4D 43 20 011D3
45 4E 4E 41 48 43 20 44 41 45 52 20 4E 55 52 011D8 P.AJD:  .ASCII  \RUN READ CHANNEL SELF-TEST\<0><0>
      00 00 54 53 45 54 2D 46 4C 45 53 20 4C 011E7
52 20 4C 4C 41 20 52 41 45 4C 43 20 4E 55 52 011F4 P.AJE:  .ASCII  \RUN CLEAR ALL RMC TEST\<0><0>
                        00 00 54 53 45 54 20 43 4D 01203
45 54 2D 46 4C 45 53 20 43 43 45 20 4E 55 52 0120C P.AJF:  .ASCII  \RUN ECC SELF-TEST\<0><0><0>
                              00 00 00 54 53 0121B
                  50 41 47 20 44 4E 49 46 01220 P.AJG:  .ASCII  \FIND GAP\
20 45 44 55 54 49 4C 50 4D 41 20 48 41 45 57 01228 P.AJH:  .ASCII  \WEAK AMPLITUDE ON PARITY BIT\
      54 49 42 20 59 54 49 52 41 50 20 4E 4F 01237
4F 4E 20 30 35 39 38 4D 20 59 54 49 52 41 50 01244 P.AJI:  .ASCII  \PARITY M8950 NOT DONE\<0><0><0>
                        00 00 00 45 4E 4F 44 20 54 01253
20 34 2D 4F 54 2D 35 20 4C 41 47 45 4C 4C 49 0125C P.AJJ:  .ASCII  \ILLEGAL 5-TO-4 FOR PARITY BIT\<0><0>
00 54 49 42 20 59 54 49 52 41 50 20 52 4F 46 0126B
                                                 00 0127A
                                                 00 0127B          .ASCII  <0>
49 52 41 50 20 52 4F 46 20 32 20 4B 52 41 4D 0127C P.AJK:  .ASCII  \MARK 2 FOR PARITY BIT\<0><0><0>
                        00 00 00 54 49 42 20 59 54 0128B
41 50 20 52 4F 46 20 4B 52 41 4D 20 44 4E 45 01294 P.AJL:  .ASCII  \END MARK FOR PARITY BIT\<0>
                        00 54 49 42 20 59 54 49 52 012A3
4F 46 20 45 4C 42 4D 41 54 53 4F 50 20 45 50 012AC P.AJM:  .ASCII  \PE POSTAMBLE FOR PARITY BIT\<0>
      00 54 49 42 20 59 54 49 52 41 50 20 52 012BB
50 54 55 4F 20 41 54 41 44 20 30 35 39 38 4D 012C8 P.AJN:  .ASCII  \M8950 DATA OUTPUT (PARITY BIT)\<0><0>
29 54 49 42 20 59 54 49 52 41 50 28 20 54 55 012D7
                                                 00 00 012E6
4F 20 44 45 54 43 45 52 52 4F 43 20 43 43 45 012E8 P.AJO:  .ASCII  \ECC CORRECTED OUTPUT (PARITY BIT)\<0><0>
42 20 59 54 49 52 41 50 28 20 54 55 50 54 55 012F7
                              00 00 29 54 49 01306
                                                 00 0130B          .ASCII  <0>
45 50 20 43 43 45 20 4B 52 54 20 4C 47 4E 53 0130C P.AJP:  .ASCII  \SNGL TRK ECC PERFORMED ON DATA\<0><0>
41 54 41 44 20 4E 4F 20 44 45 4D 52 4F 46 52 0131B
                                                 00 00 0132A
4F 46 52 45 50 20 43 43 45 20 4B 52 54 2D 32 0132C P.AJQ:  .ASCII  \2-TRK ECC PERFORMED ON DATA\<0>
      00 41 54 41 44 20 4E 4F 20 44 45 4D 52 01338
43 20 54 4F 4E 20 44 4C 55 4F 43 20 43 43 45 01348 P.AJR:  .ASCII  \ECC COULD NOT CORRECT DATA\<0><0>
      00 00 41 54 41 44 20 54 43 45 52 52 4F 01357
54 20 4F 54 20 52 45 54 4E 49 4F 50 20 4F 4E 01364 P.AJS:  .ASCII  \NO POINTER TO TRK IN ERROR\<0><0>
      00 00 52 4F 52 52 45 20 4E 49 20 4B 52 01373
20 54 4F 4E 20 44 49 44 20 22 43 52 43 41 22 01380 P.AJT:  .ASCII  \"ACRC" DID NOT CHECK\
                              4B 43 45 48 43 0138F
```

```
20 47 4E 49 52 55 44 20 22 45 49 54 4D 41 22 01394 P.AJU:  .ASCII  \"AMTIE" DURING DATA OF RECORD\<0><0>
00 44 52 4F 43 45 52 20 46 4F 20 41 54 41 44 013A3
                                              00 013B2
                                              00 013B3          .ASCII  <0>
50 20 4D 41 52 47 4F 52 50 20 31 35 39 38 4D 013B4 P.AJV:  .ASCII  \M8951 PROGRAM PARITY ERROR\<0><0>
   00 00 52 4F 52 52 45 20 59 54 49 52 41 013C3
43 20 54 4F 4E 20 44 49 44 20 22 43 52 43 22 013D0 P.AJW:  .ASCII  \"CRC" DID NOT CHECK\<0>
                                  00 48 43 45 48 013DF
      59 54 49 52 41 50 20 45 49 54 4D 41 013E4 P.AJX:  .ASCII  \AMTIE PARITY\
      00 59 54 49 52 41 50 20 44 41 45 52 013F0 P.AJY:  .ASCII  \READ PARITY\<0>
      00 00 59 54 49 52 41 50 20 53 43 57 013FC P.AJZ:  .ASCII  \WCS PARITY\<0><0>
      00 00 52 45 54 45 4D 4F 48 43 41 54 01408 P.AKA:  .ASCII  \TACHOMETER\<0><0>
45 53 45 52 50 20 54 49 4E 55 20 45 50 41 54 01414 P.AKB:  .ASCII  \TAPE UNIT PRESENT\<0><0><0>
                                  00 00 00 54 4E 01423
20 59 54 49 52 41 50 20 44 4E 41 4D 4D 4F 43 01428 P.AKC:  .ASCII  \COMMAND PARITY ERROR\
                                  52 4F 52 52 45 01437
4F 52 54 53 20 41 54 41 44 20 45 54 49 52 57 0143C P.AKD:  .ASCII  \WRITE DATA STROBE\<0><0><0>
                                  00 00 00 45 42 0144B
45 20 59 54 49 52 41 50 20 53 55 54 41 54 53 01450 P.AKE:  .ASCII  \STATUS PARITY ERROR\<0>
                                  00 52 4F 52 52 0145F
45 53 45 52 50 20 54 4F 4E 20 45 4C 42 41 43 01464 P.AKF:  .ASCII  \CABLE NOT PRESENT\<0><0><0>
                                  00 00 00 54 4E 01473
45 20 45 53 4C 55 50 20 4C 4F 52 54 4E 4F 43 01478 P.AKG:  .ASCII  \CONTROL PULSE ERROR\<0>
                                  00 52 4F 52 52 01487
4F 52 52 45 20 45 53 4C 55 50 20 41 54 41 44 0148C P.AKH:  .ASCII  \DATA PULSE ERROR\
                                              52 0149B
20 59 54 49 52 41 50 20 4C 4F 52 54 4E 4F 43 0149C P.AKI:  .ASCII  \CONTROL PARITY ERROR\
                                  52 4F 52 52 45 014AB
      00 00 00 45 54 41 4C 20 41 54 41 44 014B0 P.AKJ:  .ASCII  \DATA LATE\<0><0><0>
45 20 59 54 49 52 41 50 20 22 4D 4F 52 43 22 014BC P.AKK:  .ASCII  \"CROM" PARITY ERROR\<0>
                                  00 52 4F 52 52 014CB
4F 43 4F 54 4F 52 50 20 31 20 4C 45 56 45 4C 014D0 P.AKL:  .ASCII  \LEVEL 1 PROTOCOL ERROR\<0><0>
                     00 00 52 4F 52 52 45 20 4C 014DF
                        4B 4C 43 57 20 43 4D 58 014E8 P.AKM:  .ASCII  \XMC WCLK\
4D 57 22 20 4F 54 20 4C 41 55 44 49 53 45 52 014F0 P.AKN:  .ASCII  \RESIDUAL TO 'WMC DR" BUS\
                     53 55 42 20 22 52 44 20 43 014FF
20 43 4D 57 22 20 4F 54 20 22 43 52 43 41 22 01508 P.AKO:  .ASCII  \"ACRC" TO 'WMC DR" BUS\<0><0>
                     00 00 53 55 42 20 22 52 44 01517
44 20 43 4D 57 22 20 4F 54 20 22 43 52 43 22 01520 P.AKP:  .ASCII  \"CRC" TO 'WMC DR" BUS\<0><0><0>
                     00 00 00 53 55 42 20 22 52 0152F
22 52 44 20 43 4D 57 22 20 4F 54 20 43 43 45 01538 P.AKQ:  .ASCII  \ECC TO 'WML  R" BUS\<0>
                                  00 53 55 42 20 01547
                        52 45 46 53 4E 41 52 54 0154C P.AKR:  .ASCII  \TRANSFER\
00 45 4E 4F 44 20 54 4F 4E 20 22 43 4D 57 22 01554 P.AKS:  .ASCII  \"WMC" NOT DONE\<0><0>
                                              00 01563
00 45 4E 4F 44 20 54 4F 4E 20 22 43 4D 58 22 01564 P.AKT:  .ASCII  \"XMC" NOT DONE\<0><0>
                                              00 01573
45 54 49 52 57 20 31 20 54 52 4F 50 20 55 54 01574 P.AKU:  .ASCII  \TU PORT 1 WRITE PATH ENABLE\<0>
   00 45 4C 42 41 4E 45 20 48 54 41 50 20 01583
45 54 49 52 57 20 30 20 54 52 4F 50 20 55 54 01590 P.AKV:  .ASCII  \TU PORT 0 WRITE PATH ENABLE\<0>
   00 45 4C 42 41 4E 45 20 48 54 41 50 20 0159F
20 44 41 45 52 20 31 20 54 52 4F 50 20 55 54 015AC P.AKW:  .ASCII  \TU PORT 1 READ PATH ENABLE\<0><0>
   00 00 45 4C 42 41 4E 45 20 48 54 41 50 015BB
20 44 41 45 52 20 30 20 54 52 4F 50 20 55 54 015C8 P.AKX:  .ASCII  \TU PORT 0 READ PATH ENABLE\<0><0>
   00 00 45 4C 42 41 4E 45 20 48 54 41 50 015D7
4D 52 45 54 20 54 4E 55 4F 43 20 45 54 59 42 015E4 P.AKY:  .ASCII  \BYTE COUNT TERMINAL COUNT\<0><0><0>
      00 00 00 54 4E 55 4F 43 20 4C 41 4E 49 015F3
```

```
49 4E 55 20 45 50 41 54 20 45 4C 47 4E 49 53 01600 P.AKZ:  .ASCII  \SINGLE TAPE UNIT PORT\<0><0><0>
                  00 00 00 54 52 4F 50 20 54 0160F
45 54 49 52 57 20 33 20 54 52 4F 50 20 55 54 01618 P.ALA:  .ASCII  \TU PORT 3 WRITE PATH ENABLE\<0>
   00 45 4C 42 31 4E 45 20 48 54 41 50 20 01627
45 54 49 52 57 20 32 20 54 52 4F 50 20 55 54 01634 P.ALB:  .ASCII  \TU PORT 2 WRITE PATH ENABLE\<0>
   00 45 4C 42 41 4E 45 20 48 54 41 50 20 01643
20 44 41 45 52 20 33 20 54 52 4F 50 20 55 54 01650 P.ALC:  .ASCII  \TU PORT 3 READ PATH ENABLE\<0><0>
   00 00 45 4C 42 41 4E 45 20 48 54 41 50 0165F
20 44 41 45 52 20 32 20 54 52 4F 50 20 55 54 0166C P.ALD:  .ASCII  \TU PORT 2 READ PATH ENABLE\<0><0>
   00 00 45 4C 42 41 4E 45 20 48 54 41 50 0167B
54 49 52 41 50 20 44 41 45 52 20 22 52 44 22 01688 P.ALE:  .ASCII  \"DR" READ PARITY ERROR\<0><0>
                  00 00 52 4F 52 52 45 20 59 01697
54 49 52 41 50 20 22 4D 4F 52 20 43 4D 57 22 016A0 P.ALF:  .ASCII  \"WMC ROM" PARITY ERROR\<0><0>
                  00 00 52 4F 52 52 45 20 59 016AF
                     00 00 00 52 4F 52 52 45 016B8 P.ALG:  .ASCII  \ERROR\<0><0><0>
59 54 49 52 41 50 20 22 44 42 4D 20 52 44 22 016C0 P.ALH:  .ASCII  \"DR MBD" PARITY ERROR\<0><0><0>
                  00 00 00 52 4F 52 52 45 20 016CF
4D 4F 52 22 20 52 4F 54 41 4C 53 4E 41 52 54 016D8 P.ALI:  .ASCII  \TRANSLATOR "ROM" PARITY ERROR\<0><0>
00 52 4F 52 52 45 20 59 54 49 52 41 50 20 22 016E7
                                          00 016F6
                                          00 016F7          .ASCII  <0>
49 52 41 50 20 45 54 49 52 57 20 22 45 50 22 016F8 P.ALJ:  .ASCII  \"PE" WRITE PARITY ERROR\<0>
                  00 52 4F 52 52 45 20 59 54 01707
      00 00 54 45 53 20 22 49 4E 49 4B 22 01710 P.ALK:  .ASCII  \"KINI" SET\<0><0>
49 47 45 52 20 41 54 41 44 20 45 54 49 52 57 0171C P.ALL:  .ASCII  \WRITE DATA REGISTER PARITY\<0><0>
   00 00 59 54 49 52 41 50 20 52 45 54 53 0172B
                  4B 4F 20 52 45 57 4F 50 01738 P.ALM:  .ASCII  \POWER OK\
45 56 49 45 43 45 52 20 22 54 52 41 53 55 22 01740 P.ALN:  .ASCII  \"USART" RECEIVER RDY (RX)\<0><0><0>
   00 00 00 29 58 52 28 20 59 44 52 20 52 0174F
4B 43 4F 4C 43 20 54 50 55 52 52 45 54 4E 49 0175C P.ALO:  .ASCII  \INTERRUPT CLOCK TIME (MCLK)\<0>
   00 29 4B 4C 43 4D 28 20 45 4D 49 54 20 0176B
00 00 47 4E 49 52 20 45 54 49 52 57 20 4F 4E 01778 P.ALP:  .ASCII  \NO WRITE RING\<0><0><0>
                                          00 01787
                              00 54 4F 45 01788 P.ALQ:  .ASCII  \EOT\<0>
                              00 54 4F 42 0178C P.ALR:  .ASCII  \BOT\<0>
                              00 53 45 50 01790 P.ALS:  .ASCII  \PES\<0>
         00 00 00 47 4E 49 44 4E 49 57 45 52 01794 P.ALT:  .ASCII  \REWINDING\<0><0><0>
                     00 00 45 4E 49 4C 4E 4F 017A0 P.ALU:  .ASCII  \ONLINE\<0><0>
                     4E 4F 20 59 44 41 45 52 017A8 P.ALV:  .ASCII  \READY ON\
                     00 00 00 59 44 41 45 52 017B0 P.ALW:  .ASCII  \READY\<0><0><0>
                              00 45 53 44 017B8 P.ALX:  .ASCII  \DSE\<0>
                              00 54 4F 4D 017BC P.ALY:  .ASCII  \MOT\<0>
                              00 52 57 4C 017C0 P.ALZ:  .ASCII  \LWR\<0>
00 00 54 49 42 49 48 4E 49 20 45 54 49 52 57 017C4 P.AMA:  .ASCII  \WRITE INHIBIT\<0><0><0>
                                          00 017D3
                     00 00 00 45 54 49 52 57 017D4 P.AMB:  .ASCII  \WRITE\<0><0><0>
                     00 45 53 52 45 56 45 52 017DC P.AMC:  .ASCII  \REVERSE\<0>
                     00 44 52 41 57 52 4F 46 017E4 P.AMD:  .ASCII  \FORWARD\<0>
         00 54 53 45 54 20 4C 41 55 4E 41 4D 017EC P.AME:  .ASCII  \MANUAL TEST\<0>
         00 52 4F 52 52 45 20 22 41 52 41 22 017F8 P.AMF:  .ASCII  \"ARA" ERROR\<0>
                              0C 43 45 50 01804 P.AMG:  .ASCII  \PEC\<0>
                     00 00 45 50 20 44 4D 43 01808 P.AMH:  .ASCII  \CMD PE\<0><0>
                              48 43 41 54 01810 P.AMI:  .ASCII  \TACH\
                     00 54 45 44 20 54 4F 45 01814 P.AMJ:  .ASCII  \EOT DET\<0>
         00 45 4C 42 41 4E 45 20 44 41 45 52 0181C P.AMK:  .ASCII  \READ ENABLE\<0>
                     00 00 00 45 54 49 52 57 01828 P.AML:  .ASCII  \WRITE\<0><0><0>
         00 34 20 54 49 42 20 45 54 49 52 57 01830 P.AMM:  .ASCII  \WRITE BIT 4\<0>
```

```
20 44 45 44 4E 45 54 58 45 20 38 37 41 54 26 0183C P.AMN:  .ASCII  \&TA78 EXTENDED DRIVE STATUS INFORMATION\
4E 49 20 53 55 54 41 54 53 20 45 56 49 52 44 0184B
                  4E 4F 49 54 41 4D 52 4F 46 0185A
                                            01863          .BLKB   1
45 56 49 52 44 20 45 50 41 54 20 53 50 49 20 01864 P.AMP:  .ASCII  \ IPS TAPE DRIVE\<0>
                                            00 01873
                                  010E000F  01874 P.AMO:  .LONG   17694735
                                  00000000' 01878          .ADDRESS P.AMP
45 42 4D 55 4E 20 54 49 4E 55 20 50 43 53 4D 0187C P.AMR:  .ASCII  \MSCP UNIT NUMBER = \<0>
                                        00 20 3D 20 52 01888
                                  010E0013  01890 P.AMQ:  .LONG   17694739
                                  00000000' 01894          .ADDRESS P.AMR
                  20 3D 20 54 4E 55 4F 43 20 50 41 47 01898 P.AMT:  .ASCII  \GAP COUNT = \
                                  010E000C  018A4 P.AMS:  .LONG   17694732
                                  00000000' 018A8          .ADDRESS P.AMT
20 4C 49 41 46 20 45 54 49 52 57 20 43 4D 52 018AC P.AMV:  .ASCII  \RMC WRITE FAIL BITS\<0>
                                        00 53 54 49 42 018BB
                                  010E0013  018C0 P.AMU:  .LONG   17694739
                                  00000000' 018C4          .ADDRESS P.AMV
4F 43 4F 52 43 49 4D 20 4E 57 4F 4E 4B 4E 55 018C8 P.AMX:  .ASCII  \UNKNOWN MICROCONTROLLER STS CODE\
4F 43 20 53 54 53 20 52 45 4C 4C 4F 52 54 4E 018D7
                                        45 44 018E6
                                  010E0020  018E8 P.AMW:  .LONG   17694752
                                  00000000' 018EC          .ADDRESS P.AMX
48 43 20 45 49 54 4D 41 3E 21 20 3C 39 33 21 018F0 P.AMZ:  .ASCII  \!39< !>AMTIE CHANNEL #!AS\<0><0><0>
      00 00 00 53 41 21 23 20 4C 45 4E 4E 41 018FF
                                  010E0019  0190C P.AMY:  .LONG   17694745
                                  00000000' 01910          .ADDRESS P.AMZ
41 46 20 30 35 39 38 4D 3E 21 20 3C 39 33 21 01914 P.ANB:  .ASCII  \!39< !>M8950 FAILURE ON CHANNEL #!AS\
45 4E 4E 41 48 43 20 4E 4F 20 45 52 55 4C 49 01923
                              53 41 21 23 20 4C 01932
                                  010E0024  01938 P.ANA:  .LONG   17694756
                                  00000000' 0193C          .ADDRESS P.ANB
4C 49 20 4C 45 4E 4E 41 48 43 20 44 41 45 52 01940 P.AND:  .ASCII  \READ CHANNEL ILLEGAL STS (CH 7:0)\<0><0>
37 20 48 43 28 20 53 54 53 20 4C 41 47 45 4C 0194F
                                  00 00 29 30 3A 0195E
                                        00 01963          .ASCII  <0>
                                  010E0021  01964 P.ANC:  .LONG   17694753
                                  00000000' 01968          .ADDRESS P.AND
43 20 2C 52 4F 52 52 45 20 32 20 4B 52 41 4D 0196C P.ANF:  .ASCII  \MARK 2 ERROR, CHANNEL !ZB (M8950)\<0><0>
39 38 4D 28 20 42 5A 21 20 4C 45 4E 4E 41 48 0197B
                                  00 00 29 30 35 0198A
                                        00 0198F          .ASCII  <0>
                                  010E0021  01990 P.ANE:  .LONG   17694753
                                  00000000' 01994          .ADDRESS P.ANF
45 52 20 52 4F 46 20 4B 52 41 4D 20 44 4E 45 01998 P.ANH:  .ASCII  \END MARK FOR READ CHANNELS 7:0\<0><0>
30 3A 37 20 53 4C 45 4E 4E 41 48 43 20 44 41 019A7
                                        00 00 019B6
                                  010E001E  019B8 P.ANG:  .LONG   17694750
                                  00000000' 019BC          .ADDRESS P.ANH
45 50 20 4C 45 4E 4E 41 48 43 20 44 41 45 52 019C0 P.ANJ:  .ASCII  \READ CHANNEL PE POSTAMBLE DETECT\
45 54 45 44 20 45 4C 42 4D 41 54 53 4F 50 20 019CF
                                        54 43 019DE
                                  010E0020  019E0 P.ANI:  .LONG   17694752
                                  00000000' 019E4          .ADDRESS P.ANJ
20 44 41 45 52 20 4D 4F 52 46 20 41 54 41 44 019E8 P.ANL:  .ASCII  \DATA FROM READ CHANNELS TO ECC\<0><0>
43 43 45 20 4F 54 20 53 4C 45 4E 4E 41 48 43 019F7
```

```
                                                                00  00  01A06
                                                            010E001E  01A08 P.ANK:   .LONG   17694750
                                                            00000000'  01A0C         .ADDRESS P.ANL
54  55  4F  20  52  45  4B  43  45  48  43  20  43  52  43  01A10 P.ANN:   .ASCII  \CRC CHECKER OUTPUT BITS\<0>
                        00  53  54  49  42  20  54  55  50  01A1F
                                                            010E0017  01A28 P.ANM:   .LONG   17694743
                                                            00000000'  01A2C         .ADDRESS P.ANN
20  41  54  41  44  20  44  45  54  43  45  52  52  4F  43  01A30 P.ANP:   .ASCII  \CORRECTED DATA (ECC TO CRC)\<0>
            00  29  43  52  43  20  4F  54  20  43  43  45  28  01A3F
                                                            010E001B  01A4C P.ANO:   .LONG   17694747
                                                            00000000'  01A50         .ADDRESS P.ANP
20  45  49  54  20  50  48  43  3E  21  20  3C  39  33  21  01A54 P.ANR:   .ASCII  \!39< !>CHP TIE BUS !AS\<0><0>
                        00  00  53  41  21  20  53  55  42  01A63
                                                            010E0016  01A6C P.ANQ:   .LONG   17694742
                                                            00000000'  01A70         .ADDRESS P.ANR
20  53  55  42  20  45  49  54  3E  21  20  3C  39  33  21  01A74 P.ANT:   .ASCII  \!39< !>TIE BUS = !XB(X)\<0>
                        00  29  58  28  42  58  21  20  3D  01A83
                                                            010E0017  01A8C P.ANS:   .LONG   17694743
                                                            00000000'  01A90         .ADDRESS P.ANT
4C  20  53  55  42  20  54  49  4E  55  20  45  50  41  54  01A94 P.ANV:   .ASCII  \TAPE UNIT BUS LINE AMTIE 7:0\
            30  3A  37  20  45  49  54  4D  41  20  45  4E  49  01AA3
                                                            010E001C  01AB0 P.ANU:   .LONG   17694748
                                                            00000000'  01AB4         .ADDRESS P.ANV
41  45  52  20  45  4E  49  4C  20  53  55  42  20  55  54  01AB8 P.ANX:   .ASCII  \TU BUS LINE READ DATA 7:0\<0><0><U>
            00  00  00  30  3A  37  20  41  54  41  44  20  44  01AC7
                                                            010E0019  01AD4 P.ANW:   .LONG   17694745
                                                            00000000'  01AD8         .ADDRESS P.ANX
52  45  54  4E  49  20  2C  41  54  41  44  20  57  2F  52  01ADC P.ANZ:   .ASCII  \R/W DATA, INTERMEDIATE DRD BUS\<0><0>
53  55  42  20  44  52  44  20  45  54  41  49  44  45  4D  01AEB
                                                        00  00  01AFA
                                                            010E001E  01AFC P.ANY:   .LONG   17694750
                                                            00000000'  01B00         .ADDRESS P.ANZ
55  4F  43  20  45  54  59  42  3E  21  20  3C  39  33  21  01B04 P.AOB:   .ASCII  \!39< !>BYTE COUNT = !ZW.\
                        2E  57  5A  21  20  3D  20  54  4E  01B13
                                                            010E0018  01B1C P.AOA:   .LONG   17694744
                                                            00000000'  01B20         .ADDRESS P.AOB
4E  55  4F  43  20  44  41  50  3E  21  20  3C  39  33  21  01B24 P.AOD:   .ASCII  \!39< !>PAD COUNTER = !ZW.\<0><0><0>
            00  00  00  2E  57  5A  21  20  3D  20  52  45  54  01B33
                                                            010E0019  01B40 P.AOC:   .LONG   17694745
                                                            00000000'  01B44         .ADDRESS P.AOD
50  4D  4F  43  20  4C  55  46  53  53  45  43  43  55  53  01B48 P.AOF:   .ASCII  \SUCCESSFUL COMPLETION\<0><0><0>
                        00  00  00  4E  4F  49  54  45  4C  01B57
                                                            010E0015  01B60 P.AOE:   .LONG   17694741
                                                            00000000'  01B64         .ADDRESS P.AOF
54  4E  55  4F  43  20  50  49  4B  53  20  44  41  45  52  01B68 P.AOH:   .ASCII  \READ SKIP COUNT WAS -\<0><0><0>
                        00  00  00  2D  20  53  41  57  20  01B77
                                                            010E0015  01B80 P.AOG:   .LONG   17694741
                                                            00000000'  01B84         .ADDRESS P.AOH
55  44  20  45  52  4F  43  20  30  20  4E  49  20  34  3E  01B88 P.AOJ:   .ASCII  \>4 IN 0 CORE DUMP OR\
                                        52  4F  20  50  4D  01B97
                                                            010E0014  01B9C P.AOI:   .LONG   17694740
                                                            00000000'  01BA0         .ADDRESS P.AOJ
54  41  50  4D  4F  43  20  30  31  20  4E  49  20  33  3E  01BA4 P.AOL:   .ASCII  \>3 IN 10 COMPATIBLE OR \<0>
                        00  20  52  4F  20  45  4C  42  49  01BB3
                                                            010E0017  01BBC P.AOK:   .LONG   17694743
                                                            00000000'  01BC0         .ADDRESS P.AOL
44  2D  48  47  49  48  20  30  31  20  4E  49  20  38  3E  01BC4 P.AON:   .ASCII  \>8 IN 10 HIGH-DENSITY COMPATIBLE\
```

```
42 49 54 41 50 4D 4F 43 20 59 54 49 53 4E 45  01BD3
                                        45 4C  01BE2
                                  010E0020     01BE4 P.AOM:  .LONG   17694752
                                  00000000'    01BE8         .ADDRESS P.AON
39 33 21 2F 21 53 41 21 3E 21 20 3C 39 33 21  01BEC P.AOP:  .ASCII  \!39< !>!AS!/!39< !>!AS!/!39< !>!AS!/!39<\
21 20 3C 39 33 21 2F 21 53 41 21 3E 21 20 3C  01BFB
                  3C 39 33 21 2F 21 53 41 21  3E  01C0A
                        00 00 53 41 21 3E 21  20  01C14         .ASCII  \ !>!AS\<0><0>
                                  010E002E     01C1C P.AOO:  .LONG   17694766
                                  00000000'    01C20         .ADDRESS P.AOP
54 4E 55 4F 43 20 50 49 4B 53 20 44 41 45 52  01C24 P.AOR:  .ASCII  \READ SKIP COUNT >1 IN\<0><0><0>
                  00 00 00 4E 49 20 31 3E 20  01C33
                                  010E0015     01C3C P.AOQ:  .LONG   17694741
                                  00000000'    01C40         .ADDRESS P.AOR
4C 41 4D 52 4F 4E 20 35 31 20 72 6F 20 31 31  01C44 P.AOT:  .ASCII  \11 or 15 NORMAL\<0>
                                        00  01C53
                                  010E000F     01C54 P.AOS:  .LONG   17694735
                                  00000000'    01C58         .ADDRESS P.AOT
39 33 21 2F 21 53 41 21 3E 21 20 3C 39 33 21  01C5C P.AOV:  .ASCII  \!39< !>!AS!/!39< !>!AS\<0><0>
                  00 00 53 41 21 3E 21 20 3C  01C68
                                  010E0016     01C74 P.AOU:  .LONG   1769474
                                  00000000'    01C78         .ADDRESS P.AOV
00 36 3E 20 45 44 4F 43 20 54 41 4D 52 4F 46  01C7C P.AOX:  .ASCII  \FORMAT CODE >6\<0><0>
                                        00  01C88
                                  010E000E     01C8C P.AOW:  .LONG   17694734
                                  00000000'    01C90         .ADDRESS P.AOX
54 53 45 54 2D 46 4C 45 53 20 22 43 4D 57 22  01C94 P.AOZ:  .ASCII  \"WMC" SELF-TEST DIAGNOSTIC ERROR\
52 52 45 20 43 49 54 53 4F 4E 47 41 49 44 20  01CA3
                                        52 4F  01CB2
                                  010E0020     01CB4 P.AOY:  .LONG   17694752
                                  00000000'    01CB8         .ADDRESS P.AOZ
52 4F 20 4E 55 52 52 45 56 4F 20 44 41 45 52  01CBC P.APB:  .ASCII  \READ OVERRUN OR WRITE FAULT\<0>
      00 54 4C 55 41 46 20 45 54 49 52 57 20  01CCB
                                  010E001B     01CD8 P.APA:  .LONG   17694747
                                  00000000'    01CDC         .ADDRESS P.APB
43 20 52 4F 52 52 45 20 4E 57 4F 4E 4B 4E 55  01CE0 P.APD:  .ASCII  \UNKNOWN ERROR CODE\<0><0>
                        00 00 45 44 4F  01CEF
                                  010E0012     01CF4 P.APC:  .LONG   17694738
                                  00000000'    01CF8         .ADDRESS P.APD
49 4D 53 4E 41 52 54 20 22 54 52 41 53 55 22  01CFC P.APF:  .ASCII  \"USART" TRANSMITTER RDY (TX)\
      29 58 54 28 20 59 44 52 20 52 45 54 54  01D08
                                  010E001C     01D18 P.APE:  .LONG   17694748
                                  00000000'    01D1C         .ADDRESS P.APF
49 41 4D 28 20 32 20 4E 4F 49 54 49 53 4F 50  01D20 P.APH:  .ASCII  \POSITION 2 (MAINT)\<0><0>
                        00 00 29 54 4E  01D2F
                                  010E0012     01D34 P.APG:  .LONG   17694738
                                  00000000'    01D38         .ADDRESS P.APH
49 41 4D 28 20 31 20 4E 4F 49 54 49 53 4F 50  01D3C P.APJ:  .ASCII  \POSITION 1 (MAINT)\<0><0>
                        00 00 29 54 4E  01D4B
                                  010E0012     01D50 P.API:  .LONG   17694738
                                  00000000'    01D54         .ADDRESS P.APJ
52 4F 4E 28 20 30 20 4E 4F 49 54 49 53 4F 50  01D58 P.APL:  .ASCII  \POSITION 0 (NORMAL)\<0>
                        00 29 4C 41 4D  01D67
                                  010E0013     01D6C P.APK:  .LONG   17694739
                                  00000000'    01D70         .ADDRESS P.APL
49 41 4D 28 20 33 20 4E 4F 49 54 49 53 4F 50  01D74 P.APN:  .ASCII  \POSITION 3 (MAINT)\<0><0>
                        00 00 29 54 4E  01D83
```

```
                                                  010E0012  01D88 P.APM:   .LONG   17694738                                ;
                                                  00000000' 01D8C          .ADDRESS P.APN                                  ;
45 53 20 45 44 4F 4D 20 4E 57 4F 4E 4B 4E 55      01D90 P.APP:   .ASCII  \UNKNOWN MODE SELECTION\<0><0>               ;
                  00 00 4E 4F 49 54 43 45 4C      01D9F                                                                 ;
                                                  010E0016  01DA8 P.APO:   .LONG   17694742                                ;
                                                  00000000' 01DAC          .ADDRESS P.APP                                  ;
52 44 20 45 50 41 54 20 53 50 4? 20 35 32 31      01DB0 P.APR:   .ASCII  \125 IPS TAPE DRIVE\<0><0>                   ;
                              00 00 45 56 49      01DBF                                                                 ;
                                                  010E0012  01DC4 P.APQ:   .LONG   17694738                                ;
                                                  00000000' 01DC8          .ADDRESS P.APR                                  ;
41 49 52 45 53 20 54 49 4E 55 20 45 50 41 54      01DCC P.APT:   .ASCII  \TAPE UNIT SERIAL #\<0><0>                   ;
                              00 00 23 20 4C      01DDB                                                                 ;
                                                  010E0012  01DE0 P.APS:   .LONG   17694738                                ;
                                                  00000000' 01DE4          .ADDRESS P.APT                                  ;
44 4C 4F 48 53 45 52 48 54 20 45 49 54 4D 41      01DE8 P.APV:   .ASCII  \AMTIE THRESHOLD FIELD = \                   ;
                  20 3D 20 44 4C 45 49 46 20      01DF7                                                                 ;
                                                  010E0018  01E00 P.APU:   .LONG   17694744                                ;
                                                  00000000' 01E04          .ADDRESS P.APV                                  ;

                                                                   .PSECT  $OWN$,NOEXE,  PIC,2

                                 35 2D 16 0E 07  002B4 B1_DIAG_STS_CODES:
                                                                   .BYTE   7, 14, 22, 45, 53                               ;
                                                 002B9             .BLKB   3
                           15 0D 08 04 03 02 01  002BC B2_DIAG_STS_CODES:
                                                                   .BYTE   1, 2, 3, 4, 8, 13, 21                           ;
                                                 002C3             .BLKB   1
9C 99 98 92 90 8A 89 88 81 46 44 43 42 41 01     002C4 RMC_STS_CODES:
                                                                   .BYTE   1, 65, 66, 67, 68, 70, -127, -120, -119, -    ;
                              FF B2 B1 A1 9E 9D  002D3                     -118, -112, -110, -104, -103, -100, -99, -    ;
                                                                           -98, -95, -79, -78, -1                          ;
                                                 002D9             .BLKB   3
                                      00000000'  002DC TU_STRING:
                                                                   .ADDRESS P.AGX                                  ;
                                                 002E0 SERIAL_NUMBER:
                                                                   .BLKB   2
                                                 002E2             .BLKB   2
                                      0000000D   002E4 B1_DIAG_STS_CODES_DESC_TBL:
                                                                   .LONG   13                                      ;
                                      00000000'  002E8             .ADDRESS P.AGZ                                  ;
                                      0000000D   002EC             .LONG   13                                      ;
                                      00000000'  002F0             .ADDRESS P.AHA                                  ;
                                      00000013   002F4             .LONG   19                                      ;
                                      00000000'  002F8             .ADDRESS P.AHB                                  ;
                                      00000013   002FC             .LONG   19                                      ;
                                      00000000'  00300             .ADDRESS P.AHC                                  ;
                                      00000019   00304             .LONG   25                                      ;
                                      00000000'  00308             .ADDRESS P.AHD                                  ;
                                      00000019   0030C B2_DIAG_STS_CODES_DESC_TBL:
                                                                   .LONG   25                                      ;
                                      00000000'  00310             .ADDRESS P.AHE                                  ;
                                      0000001F   00314             .LONG   31                                      ;
                                      00000000'  00318             .ADDRESS P.AHF                                  ;
                                      0000001A   0031C             .LONG   26                                      ;
                                      00000000'  00320             .ADDRESS P.AHG                                  ;
                                      00000020   00324             .LONG   32                                      ;
                                      00000000'  00328             .ADDRESS P.AHH                                  ;
```

```
00000005  0032C                .LONG   5
00000000' 00330                .ADDRESS P.AHI
0000000C  00334                .LONG   12
00000000' 00338                .ADDRESS P.AHJ
00000012  0033C                .LONG   18
00000000  00340                .ADDRESS P.AHK
00000015  00344 BYTE2_DESC_TBL:
                               .LONG   21
00000000' 00348                .ADDRESS P.AHL
00000014  0034C                .LONG   20
00000000' 00350                .ADDRESS P.AHM
00000015  00354                .LONG   21
00000000' 00358                .ADDRESS P.AHN
0000000C  0035C BYTE17_DESC_TBL:
                               .LONG   12
00000000' 00360                .ADDRESS P.AHO
00000011  00364                .LONG   17
00000000' 00368                .ADDRESS P.AHP
0000000D  0036C                .LONG   13
00000000' 00370                .ADDRESS P.AHQ
00000015  00374                .LONG   21
00000000' 00378                .ADDRESS P.AHR
0000000A  0037C                .LONG   10
00000000' 00380                .ADDRESS P.AHS
0000000E  00384                .LONG   14
00000000' 00388                .ADDRESS P.AHT
0000000C  0038C                .LONG   12
00000000' 00390                .ADDRESS P.AHU
0000000B  00394                .LONG   11
00000000' 00398                .ADDRESS P.AHV
00000010  0039C BYTE18_DESC_TBL:
                               .LONG   16
00000000' 003A0                .ADDRESS P.AHW
0000001F  003A4                .LONG   31
00000000' 003A8                .ADDRESS P.AHX
0000001F  003AC                .LONG   31
00000000' 003B0                .ADDRESS P.AHY
0000001A  003B4                .LONG   26
00000000' 003B8                .ADDRESS P.AHZ
0000001A  003BC                .LONG   26
00000000' 003C0                .ADDRESS P.AIA
0000001D  003C4                .LONG   29
00000000' 003C8                .ADDRESS P.AIB
0000000F  003CC                .LONG   15
00000000' 003D0                .ADDRESS P.AIC
00000013  003D4                .LONG   19
00000000' 003D8                .ADDRESS P.AID
0000000E  003DC                .LONG   14
00000000' 003E0                .ADDRESS P.AIE
0000000D  003E4                .LONG   13
00000000' 003E8                .ADDRESS P.AIF
00000017  003EC                .LONG   23
00000000' 003F0                .ADDRESS P.AIG
00000020  003F4                .LONG   32
00000000' 003F8                .ADDRESS P.AIH
0000000E  003FC                .LONG   14
00000000' 00400                .ADDRESS P.AII
```

```
0000000F  00404          .LONG   15
00000000' 00408          .ADDRESS P.AIJ
00000016  0040C          .LONG   22
00000000' 00410          .ADDRESS P.AIK
00000011  00414          .LONG   17
00000000' 00418          .ADDRESS P.AIL
00000011  0041C          .LONG   17
00000000' 00420          .ADDRESS P.AIM
00000018  00424          .LONG   24
00000000' 00428          .ADDRESS P.AIN
0000000E  0042C          .LONG   14
00000000' 00430          .ADDRESS P.AIO
0000000F  00434          .LONG   15
00000000' 00438          .ADDRESS P.AIP
00000002  0043C          .LONG   2
00000000' 00440          .ADDRESS P.AIQ
00000003  00444 BYTE19_DESC_TBL:
                         .LONG   3
00000000' 00448          .ADDRESS P.AIR
0000000F  0044C          .LONG   15
00000000' 00450          .ADDRESS P.AIS
00000010  00454          .LONG   16
00000000' 00458          .ADDRESS P.AIT
00000011  0045C          .LONG   17
00000000' 00460          .ADDRESS P.AIU
00000011  00464          .LONG   17
00000000' 00468          .ADDRESS P.AIV
0000000E  0046C          .LONG   14
00000000' 00470          .ADDRESS P.AIW
0000000E  00474          .LONG   14
00000000' 00478          .ADDRESS P.AIX
00000013  0047C          .LONG   19
00000000' 00480          .ADDRESS P.AIY
00000011  00484          .LONG   17
00000000' 00488          .ADDRESS P.AIZ
00000015  0048C          .LONG   21
00000000' 00490          .ADDRESS P.AJA
0000001C  00494          .LONG   28
00000000' 00498          .ADDRESS P.AJB
00000013  0049C          .LONG   19
00000000' 004A0          .ADDRESS P.AJC
0000001A  004A4          .LONG   26
00000000' 004A8          .ADDRESS P.AJD
00000016  004AC          .LONG   22
00000000' 004B0          .ADDRESS P.AJE
00000011  004B4          .LONG   17
00000000' 004B8          .ADDRESS P.AJF
00000008  004BC          .LONG   8
00000000' 004C0          .ADDRESS P.AJG
0000001C  004C4 BYTE25_DESC_TBL:
                         .LONG   28
00000000' 004C8          .ADDRESS P.AJH
00000015  004CC          .LONG   21
00000000' 004D0          .ADDRESS P.AJI
0000001D  004D4          .LONG   29
00000000' 004D8          .ADDRESS P.AJJ
00000015  004DC          .LONG   21
```

```
00000000' 004E0          .ADDRESS P.AJK
00000017  004E4          .LONG   23
00000000' 004E8          .ADDRESS P.AJL
0000001B  004EC          .LONG   27
00000000' 004F0          .ADDRESS P.AJM
0000001E  004F4          .LONG   30
00000000' 004F8          .ADDRESS P.AJN
00000021  004FC          .LONG   33
00000000' 00500          .ADDRESS P.AJO
0000001E  00504 BYTE30_DESC_TBL:
                         .LONG   30
00000000' 00508          .ADDRESS P.AJP
0000001B  0050C          .LONG   27
00000000' 00510          .ADDRESS P.AJQ
0000001A  00514          .LONG   26
00000000' 00518          .ADDRESS P.AJR
0000001A  0051C          .LONG   26
00000000' 00520          .ADDRESS P.AJS
00000014  00524          .LONG   20
00000000' 00528          .ADDRESS P.AJT
0000001D  0052C          .LONG   29
00000000' 00530          .ADDRESS P.AJU
0000001A  00534          .LONG   26
00000000' 00538          .ADDRESS P.AJV
00000013  0053C          .LONG   19
00000000' 00540          .ADDRESS P.AJW
0000000C  00544 BYTE37_DESC_TBL:
                         .LONG   12
00000000' 00548          .ADDRESS P.AJX
0000000B  0054C          .LONG   11
00000000' 00550          .ADDRESS P.AJY
0000000A  00554          .LONG   10
00000000' 00558          .ADDRESS P.AJZ
0000000A  0055C          .LONG   10
00000000' 00560          .ADDRESS P.AKA
00000011  00564          .LONG   17
00000000' 00568          .ADDRESS P.AKB
00000014  0056C          .LONG   20
00000000' 00570          .ADDRESS P.AKC
00000011  00574          .LONG   17
00000000' 00578          .ADDRESS P.AKD
00000013  0057C          .LONG   19
00000000' 00580          .ADDRESS P.AKE
00000011  00584 BYTE39_DESC_TBL:
                         .LONG   17
00000000' 00588          .ADDRESS P.AKF
00000013  0058C          .LONG   19
00000000' 00590          .ADDRESS P.AKG
00000010  00594          .LONG   16
00000000' 00598          .ADDRESS P.AKH
00000014  0059C          .LONG   20
00000000' 005A0          .ADDRESS P.AKI
00000009  005A4          .LONG   9
00000000' 005A8          .ADDRESS P.AKJ
00000013  005AC          .LONG   19
00000000' 005B0          .ADDRESS P.AKK
00000016  005B4          .LONG   22
```

```
00000000' 005B8          .ADDRESS P.AKL
00000008  005BC BYTE40_DESC_TBL:
                         .LONG   8
00000000' 005C0          .ADDRESS P.AKM
00000018  005C4          .LONG   24
00000000' 005C8          .ADDRESS P.AKN
00000016  005CC          .LONG   22
00000000' 005D0          .ADDRESS P.AKO
00000015  005D4          .LONG   21
00000000' 005D8          .ADDRESS P.AKP
00000013  005DC          .LONG   19
00000000' 005E0          .ADDRESS P.AKQ
00000008  005E4          .LONG   8
00000000' 005E8          .ADDRESS P.AKR
0000000E  005EC          .LONG   14
00000000' 005F0          .ADDRESS P.AKS
0000000E  005F4          .LONG   14
00000000' 005F8          .ADDRESS P.AKT
0000001B  005FC BYTE41_DESC_TBL:
                         .LONG   27
00000000' 00600          .ADDRESS P.AKU
0000001B  00604          .LONG   27
00000000' 00608          .ADDRESS P.AKV
0000001A  0060C          .LONG   26
00000000' 00610          .ADDRESS P.AKW
0000001A  00614          .LONG   26
00000000' 00618          .ADDRESS P.AKX
00000019  0061C          .LONG   25
00000000' 00620          .ADDRESS P.AKY
00000015  00624          .LONG   21
00000000' 00628          .ADDRESS P.AKZ
0000001B  0062C BYTE42_DESC_TBL:
                         .LONG   27
00000000' 00630          .ADDRESS P.ALA
0000001B  00634          .LONG   27
00000000' 00638          .ADDRESS P.ALB
0000001A  0063C          .LONG   26
00000000' 00640          .ADDRESS P.ALC
0000001A  00644          .LONG   26
00000000' 00648          .ADDRESS P.ALD
00000016  0064C BYTE50_DESC_TBL:
                         .LONG   22
00000000' 00650          .ADDRESS P.ALE
00000016  00654          .LONG   22
00000000' 00658          .ADDRESS P.ALF
00000005  0065C          .LONG   5
00000000' 00660          .ADDRESS P.ALG
00000015  00664          .LONG   21
00000000' 00668          .ADDRESS P.ALH
0000001D  0066C BYTE51_DESC_TBL:
                         .LONG   29
00000000' 00670          .ADDRESS P.ALI
00000017  00674          .LONG   23
00000000' 00678          .ADDRESS P.ALJ
0000000A  0067C BYTE51_2_DESC_TBL:
                         .LONG   10
00000000' 00680          .ADDRESS P.ALK
```

```
0000001A  00684          .LONG   26
00000000' 00688          .ADDRESS P.ALL
00000008  0068C          .LONG   8
00000000' 00690          .ADDRESS P.ALM
00000019  00694 BYTE51_3_DESC_TBL:
                         .LONG   25
00000000' 00698          .ADDRESS P.ALN
0000001B  0069C          .LONG   27
00000000' 006A0          .ADDRESS P.ALO
0000000D  006A4 BYTE52_DESC_TBL:
                         .LONG   13
00000000' 006A8          .ADDRESS P.ALP
00000003  006AC          .LONG   3
00000000' 006B0          .ADDRESS P.ALQ
00000003  006B4          .LONG   3
00000000' 006B8          .ADDRESS P.ALR
00000003  006BC          .LONG   3
00000000' 006C0          .ADDRESS P.ALS
00000009  006C4          .LONG   9
00000000' 006C8          .ADDRESS P.ALT
00000006  006CC          .LONG   6
00000000' 006D0          .ADDRESS P.ALU
00000008  006D4          .LONG   8
00000000' 006D8          .ADDRESS P.ALV
00000005  006DC          .LONG   5
00000000' 006E0          .ADDRESS P.ALW
00000003  006E4 BYTE53_DESC_TBL:
                         .LONG   3
00000000' 006E8          .ADDRESS P.ALX
00000003  006EC          .LONG   3
00000000' 006F0          .ADDRESS P.ALY
00000003  006F4          .LONG   3
00000000' 006F8          .ADDRESS P.ALZ
0000000D  006FC          .LONG   13
00000000' 00700          .ADDRESS P.AMA
00000005  00704          .LONG   5
00000000' 00708          .ADDRESS P.AMB
00000007  0070C          .LONG   7
00000000' 00710          .ADDRESS P.AMC
00000007  00714          .LONG   7
00000000' 00718          .ADDRESS P.AMD
0000000B  0071C          .LONG   11
00000000' 00720          .ADDRESS P.AME
0000000B  00724 BYTE54_DESC_TBL:
                         .LONG   11
00000000' 00728          .ADDRESS P.AMF
00000003  0072C          .LONG   3
00000000' 00730          .ADDRESS P.AMG
00000006  00734          .LONG   6
00000000' 00738          .ADDRESS P.AMH
00000004  0073C BYTE57_DESC_TBL:
                         .LONG   4
00000000' 00740          .ADDRESS P.AMI
00000007  00744          .LONG   7
00000000' 00748          .ADDRESS P.AMJ
0000000B  0074C          .LONG   11
00000000' 00750          .ADDRESS P.AMK
```

```
                                00000005  00754          .LONG   5                                                       ;
                                00000000' 00758          .ADDRESS P.AML                                                  ;
                                0000000B  0075C          .LONG   11                                                      ;
                                00000000' 00760          .ADDRESS P.AMM                                                  ;



                                                         .PSECT  $CODE,NOWRT,  PIC,2

                                          0FFC 00000     .ENTRY  TA78_DRIVE_STS_DECODE, Save R2,R3,R4,R5,R6,-;  1250
                                                                 R7,R8,R9,RT0,RT1                                        ;
                            5B 00000000G  00   9E 00002  MOVAB   TRANSLATE_BITS, R11                                     ;
                            5A 00000000'  00   9E 00009  MOVAB   P.AMN, R10                                              ;
                            59 00000000G  00   9E 00010  MOVAB   DRIVE_STS+50, R9                                        ;
                            58 00000000V  00   9E 00017  MOVAB   LABEL_AND_HEX_OUTPUT, R8                                ;
                            57 00000000G  00   9E 0001E  MOVAB   OUTPUT_LINES, R7                                        ;
                            56 00000000'  00   9E 00025  MOVAB   ARGLIST, R6                                             ;
                            5E            04   C2 0002C  SUBL2   #4, SP                                                ;
                  55    03  A9            03   00 EF 0002F  EXTZV   #0, #3, DRIVE_STS+53, R5                            ;  1484
                            54        05  A9   9A 00035  MOVZBL  DRIVE_STS+55, R4                                       ;  1486
                            53        04  A9   9A 00039  MOVZBL  DRIVE_STS+54, R3                                       ;  1487
                            66            6A   9E 0003D  MOVAB   P.AMN, ARGLIST                                        ;  1492
                       00A0 C6            01   90 00040  MOVB    #1, MAIN_HDR                                          ;  1493
                                          56   DD 00045  PUSHL   R6                                                    ;  1494
                                      68  A6   DD 00047  PUSHL   FAO_STRINGS                                           ;
                            67            02   FB 0004A  CALLS   #2, OUTPUT_LINES                                      ;
                                          01   DD 0004D  PUSHL   #1                                                    ;  1499
                            68            01   FB 0004F  CALLS   #1, LABEL_AND_HEX_OUTPUT                              ;
                            66        CE  A9   9A 00052  MOVZBL  DRIVE_STS, ARGLIST                                    ;  1501
                         04 A6        38  AA   9E 00056  MOVAB   P.AMO, ARGLIST+4                                     ;  1502
                                          56   DD 0005B  PUSHL   R6                                                    ;  1503
                                  0094    C6   DD 0005D  PUSHL   FAO_STRINGS+44                                        ;
                            67            02   FB 00061  CALLS   #2, OUTPUT_LINES                                      ;
                                          02   DD 00064  PUSHL   #2                                                    ;  1508
                            68            01   FB 00066  CALLS   #1, LABEL_AND_HEX_OUTPUT                              ;
                                          6E   D4 00069  CLRL    (SP)                                                  ;  1511
                                          5E   DD 0006B  PUSHL   SP                                                    ;
                                       64 A6   9F 0006D  PUSHAB  FAO_STRING                                            ;  1510
                                  00A4 C6      9F 00070  PUSHAB  OUT_ARGLIST                                           ;
                                  0344 C6      9F 00074  PUSHAB  BYTE2_DESC_TBL                                        ;
                                       5C A6   9F 00078  PUSHAB  BIT_MSK_11                                            ;
                                       CF A9   9F 0007B  PUSHAB  DRIVE_STS+1                                           ;
                            6B            06   FB 0007E  CALLS   #6, TRANSLATE_BITS                                   ;
                            0A            50   E9 00081  BLBC    R0, 1$                                                ;
                                  00A4    C6   DD 00084  PUSHL   OUT_ARGLIST                                           ;  151?
                                       64 A6   DD 00088  PUSHL   FAO_STRING                                            ;
                            67            02   FB 0008B  CALLS   #2, OUTPUT_LINES                                      ;
                                          04   DD 0008E  1$:    PUSHL   #4                                            ;  1519
                                          03   DD 00090  PUSHL   #3                                                    ;
                            68            02   FB 00092  CALLS   #2, LABEL_AND_HEX_OUTPUT                              ;
                            66        54  AA   9E 00095  MOVAB   P.AMQ, ARGLIST                                        ;  1521
                         04 A6        D0  A9   3C 00099  MOVZWL  UNIT_NUMBER, ARGLIST+4                                ;  1522
                                          56   DD 0009E  PUSHL   R6                                                    ;  1524
                                       70 A6   DD 000A0  PUSHL   FAO_STRINGS+8                                         ;
                            67            02   FB 000A3  CALLS   #2, OUTPUT_LINES                                      ;
                                          08   DD 000A6  PUSHL   #8                                                    ;  1529
                                          07   DD 000A8  PUSHL   #7                                                    ;
```

```
                                 06  DD 000AA          PUSHL   #6                                          ;
                                 05  DD 000AC          PUSHL   #5                                          ;
                             68  04  FB 000AE          CALLS   #4, LABEL_AND_HEX_OUTPUT                    ;
                             66      68  AA  9E 000B1  MOVAB   P.AMS, ARGLIST                             ; 1531
                          04 A6      D2  A9  D0 000B5  MOVL    GAP_COUNT, ARGLIST+4                      ; 1532
                                 56  DD 000BA          PUSHL   R6                                          ; 1534
                                     70  A6  DD 000BC  PUSHL   FAO_STRINGS+8
                             67      02  FB 000BF      CALLS   #2, OUTPUT_LINES                            ;
                                 09  DD 000C2          PUSHL   #9                                          ; 1540
                             68      01  FB 000C4      CALLS   #1, LABEL_AND_HEX_OUTPUT                    ;
                                   02E4  C6  9F 000C7  PUSHAB  B1_DIAG_STS_CODES_DESC_TBL                  ; 1541
                                 05  DD 000CB          PUSHL   #5                                          ;
                                   02B4  C6  9F 000CD  PUSHAB  B1_DIAG_STS_CODES
                                 01  DD 000D1          PUSHL   #1                                          ;
                  00000000V      00  04  FB 000D3      CALLS   #4, DIAGNOSTIC_STS_BYTE_DECODE              ;
                                 0A  DD 000DA          PUSHL   #10                                         ; 1547
                             68      01  FB 000DC      CALLS   #1, LABEL_AND_HEX_OUTPUT                    ;
                                   030C  C6  9F 000DF  PUSHAB  B2_DIAG_STS_CODES_DESC_TBL                  ; 1548
                                 07  DD 000E3          PUSHL   #7                                          ;
                                   02BC  C6  9F 000E5  PUSHAB  B2_DIAG_STS_CODES
                                 02  DD 000E9          PUSHL   #2                                          ;
                  00000000V      00  04  FB 000EB      CALLS   #4, DIAGNOSTIC_STS_BYTE_DECODE              ;
                                     D6  A9  95 000F2  TSTB    DRIVE_STS+8                               ; 1556
                                 03  12  000F5          BNEQ    2$                                          ;
                                     D7  A9  95 000F7  TSTB    DRIVE_STS+9                               ; 1557
                                 01  13  000FA  2$:     BEQL    3$                                          ;
                                     04  0C0FC          RET                                                 ;
                                 0C  DD 000FD  3$:      PUSHL   #12                                         ; 1564
                                 0B  DD 000FF          PUSHL   #11                                         ;
                             68  02  FB 00101          CALLS   #2, LABEL_AND_HEX_OUTPUT                    ;
                                 0D  DD 00104          PUSHL   #13                                         ; 1569
                             68  01  FB 00106          CALLS   #1, LABEL_AND_HEX_OUTPUT                    ;
                             66      00A8  C6  D0 00109  MOVL    STI_STRING, ARGLIST                       ; 1571
                          04 A6      DA  A9  9A 0010E  MOVZBL  DRIVE_STS+12, ARGLIST+4                   ; 1572
                                 56  DD 00113          PUSHL   R6                                          ; 1574
                                   0084  C6  DD 00115  PUSHL   FAO_STRINGS+28
                             67      02  FB 00119      CALLS   #2, OUTPUT_LINES                            ;
                                 0F  DD 0011C          PUSHL   #15                                         ; 1580
                                 0E  DD 0011E          PUSHL   #14                                         ;
                             68  02  FB 00120          CALLS   #2, LABEL_AND_HEX_OUTPUT                    ;
                             7E      DB  A9  3C 00123  MOVZWL  ERRNUM, -(SP)                               ; 1581
                        FE16 CF      01  FB 00127      CALLS   #1, ERROR_NUMBER_BYTE_DECODE                ;
                                 10  DD 0012C          PUSHL   #16                                         ; 1586
                             68  01  FB 0012E          CALLS   #1, LABEL_AND_HEX_OUTPUT                    ;
                             66      0084  CA  9E 00131  MOVAB   P.AMU, ARGLIST                            ; 1587
                                 56  DD 00136          PUSHL   R6                                          ; 1588
                                     74  A6  DD 00138  PUSHL   FAO_STRINGS+12
                             67      02  FB 0013B      CALLS   #2, OUTPUT_LINES                            ;
                                 11  DD 0013E          PUSHL   #17                                         ; 1593
                             68  01  FB 00140          CALLS   #1, LABEL_AND_HEX_OUTPUT                    ;
                                 6E  D4 00143          CLRL    (SP)                                        ; 1596
                                 5E  DD 00145          PUSHL   SP                                          ;
                                     64  A6  9F 00147  PUSHAB  FAO_STRING                                 ; 1595
                                   00A4  C6  9F 0014A  PUSHAB  OUT_ARGLIST
                                   035C  C6  9F 0014E  PUSHAB  BYTE17_DESC_TBL
                                     30  A6  9F 00152  PUSHAB  BIT_MSK_0
                                     DE  A9  9F 00155  PUSHAB  DRIVE_STS+16
```

```
                       6B              06  FB 00158          CALLS   #6, TRANSLATE_BITS                     :
                       0A              50  E9 0015B          BLBC    R0, 4$                                 :
                                00A4   C6  DD 0015E          PUSHL   OUT_ARGLIST                            : 1598
                                  64   A6  DD 00162          PUSHL   FAO_STRING                             :
                       67              02  FB 00165          CALLS   #2, OUTPUT_LINES                       :
                                       12  DD 00168 4$:      PUSHL   #18                                    : 1604
                       68              01  FB 0016A          CALLS   #1, LABEL_AND_HEX_OUTPUT               :
                       66       00AC   CA  9E 0016D          MOVAB   P.AMW, ARGLIST                         : 1606
                                       50  D4 00172          CLRL    I                                      : 1610
                 02C4 C640        DF   A9  91 00174 5$:      CMPB    DRIVE_STS+17, RMC_STS_CODES[I]         :
                                       08  12 0017B          BNEQ    6$                                     :
                       66  039C C640   7E  00017D            MOVAQ   BYTE18_DESC_TBL[I], ARGLIST            : 1613
                                       04  11 00183          BRB     7$                                     : 1612
             EB        50              14  F3 00185 6$:      AOBLEQ  #20, I, 5$                             : 1608
                                       56  DD 00189 7$:      PUSHL   R6                                     : 1618
                                  74   A6  DD 0018B          PUSHL   FAO_STRINGS+12                         :
                       67              02  FB 0018E          CALLS   #2, OUTPUT_LINES                       :
                                       13  DD 00191          PUSHL   #19                                    : 1624
                       68              01  FB 00193          CALLS   #1, LABEL_AND_HEX_OUTPUT               :
                       50         E0   A9  9A 00196          MOVZBL  DRIVE_STS+18, R0                       : 1626
                       66  0444 C640   7E  00019A            MOVAQ   BYTE19_DESC_TBL[R0], ARGLIST           :
                                       56  DD 001A0          PUSHL   R6                                     : 1628
                                0090   C6  DD 001A2          PUSHL   FAO_STRINGS+40                         :
                       67              02  FB 001A6          CALLS   #2, OUTPUT_LINES                       :
                                       14  DD 001A9          PUSHL   #20                                    : 1634
                       68              01  FB 001AB          CALLS   #1, LABEL_AND_HEX_OUTPUT               :
                  64   A6       00D0   CA  9E 001AE          MOVAB   P.AMY, FAO_STRING                      : 1636
                                       52  D4 001B4          CLRL    I                                      : 1637
             0E   E1   A9              52  E1 001B6 8$:      BBC     I, DRIVE_STS+19, 9$                    : 1639
                       66  00AC C642   7E  001BB             MOVAQ   BYTE31_DESC_TBL[I], ARGLIST            : 1642
                                       56  DD 001C1          PUSHL   R6                                     : 1643
                                  64   A6  DD 001C3          PUSHL   FAO_STRING                             :
                       67              02  FB 001C6          CALLS   #2, OUTPUT_LINES                       :
             E9        52              07  F3 001C9 9$:      AOBLEQ  #7, I, 8$                              : 1637
                                       15  DD 001CD          PUSHL   #21                                    : 1652
                       68              01  FB 001CF          CALLS   #1, LABEL_AND_HEX_OUTPUT               :
                  64   A6       00FC   CA  9E 001D2          MOVAB   P.ANA, FAO_STRING                      : 1654
                                       52  D4 001D8          CLRL    I                                      : 1655
             0E   E2   A9              52  E1 001DA 10$:     BBC     I, DRIVE_STS+20, 11$                   : 1657
                       66  00AC C642   7E  001DF             MOVAQ   BYTE31_DESC_TBL[I], ARGLIST            : 1660
                                       56  DD 001E5          PUSHL   R6                                     : 1661
                                  64   A6  DD 001E7          PUSHL   FAO_STRING                             :
                       67              02  FB 001EA          CALLS   #2, OUTPUT_LINES                       :
             E9        52              07  F3 001ED 11$:     AOBLEQ  #7, I, 10$                             : 1655
                                       16  DD 001F1          PUSHL   #22                                    : 1668
                       68              01  FB 001F3          CALLS   #1, LABEL_AND_HEX_OUTPUT               :
                       66       0128   CA  9E 001F6          MOVAB   P.ANC, ARGLIST                         : 1670
                                       56  DD 001FB          PUSHL   R6                                     : 1671
                                  74   A6  DD 001FD          PUSHL   FAO_STRINGS+12                         :
                       67              02  FB 00200          CALLS   #2, OUTPUT_LINES                       :
                                       17  DD 00203          PUSHL   #23                                    : 1676
                       68              01  FB 00205          CALLS   #1, LABEL_AND_HEX_OUTPUT               :
                  64   A6       0154   CA  9E 00208          MOVAB   P.ANE, FAO_STRING                      : 1678
                                       52  D4 0020E          CLRL    I                                      : 1679
    50       E4   A9              20   52  EF 00210 12$:     EXTZV   I, #32, DRIVE_STS+22, R0               : 1681
                       0B              50  E9 00216          BLBC    R0, 13$                                :
                       66              50  D0 00219          MOVL    R0, ARGLIST                            : 1684
```

```
                        56  DD 0021C          PUSHL   R6                                          ; 1685
                     64 A6  DD 00215          PUSHL   FAO_STRING
                     67     02 FB 00221       CALLS   #2, OUTPUT_LINES
                  E8 52     07 F3 00224 13$:  ASBLEQ  #7, I, 12$                                  ; 1679
                        18  DD 00228          PUSHL   #24                                         ; 1692
                     68     01 FB 0022A       CALLS   #1, LABEL_AND_HEX_OUTPUT
                     66 017C CA 9E 0022D       MOVAB   P.ANG, ARGLIST                              ; 1694
                        56  DD 00232          PUSHL   R6                                          ; 1695
                     74 A6  DD 00234          PUSHL   FAO_STRINGS+12
                     67     02 FB 00237       CALLS   #2, OUTPUT_LINES
                        19  DD 0023A          PUSHL   #25                                         ; 1700
                     68     01 FB 0023C       CALLS   #1, LABEL_AND_HEX_OUTPUT
                        6E  D4 0023F          CLRL    (SP)                                        ; 1703
                        5E  DD 00241          PUSHL   SP
                     64 A6  9F 00243          PUSHAB  FAO_STRING                                  ; 1702
                   00A4 C6  9F 00246          PUSHAB  OUT_ARGLIST
                   04C4 C6  9F 0024A          PUSHAB  BYTE25_DESC_TBL
                     30 A6  9F 0024E          PUSHAB  BIT_MSR_0
                     E6 A9  9F 00251          PUSHAB  DRIVE_STS+24
                     6B     06 FB 00254       CALLS   #6, TRANSLATE_BITS
                     0A     50 E9 00257       BLBC    R0, 14$
                   00A4 C6  DD 0025A          PUSHL   OUT_ARGLIST                                 ; 1705
                     64 A6  DD 0025E          PUSHL   FAO_STRING
                     67     02 FB 00261       CALLS   #2, OUTPUT_LINES
                        1A  DD 00264 14$:     PUSHL   #26                                         ; 1710
                     68     01 FB 00266       CALLS   #1, LABEL_AND_HEX_OUTPUT
                     66 01A4 CA 9E 00269       MOVAB   .ANI, ARGLIST                               ; 1712
                        56  DD 0026E          PUSHL   R6                                          ; 1713
                     74 A6  DD 00270          PUSHL   FAO_STRINGS+12
                     67     02 FB 00273       CALLS   #2, OUTPUT_LINES
                        1B  DD 00276          PUSHL   #27                                         ; 1719
                     68     01 FB 00278       CALLS   #1, LABEL_AND_HEX_OUTPUT
                     66 01CC CA 9E 0027B       MOVAB   P.ANK, ARGLIST                              ; 1721
                        56  DD 00280          PUSHL   R6                                          ; 1722
                     74 A6  DD 00282          PUSHL   FAO_STRINGS+12
                     67     02 FB 00285       CALLS   #2, OUTPUT_LINES
                        1C  DD 00288          PUSHL   #28                                         ; 1727
                     68     01 FB 0028A       CALLS   #1, LABEL_AND_HEX_OUTPUT
                     66 01EC CA 9E 0028D       MOVAB   P.ANM, ARGLIST                              ; 1729
                        56  DD 00292          PUSHL   R6                                          ; 1730
                     74 A6  DD 00294          PUSHL   FAO_STRINGS+12
                     67     02 FB 00297       CALLS   #2, OUTPUT_LINES
                        1D  DD 0029A          PUSHL   #29                                         ; 1735
                     68     01 FB 0029C       CALLS   #1, LABEL_AND_HEX_OUTPUT
                     66 0210 CA 9E 0029F       MOVAB   P.ANO, ARGLIST                              ; 1737
                        56  DD 002A4          PUSHL   R6                                          ; 1738
                     74 A6  DD 002A6          PUSHL   FAO_STRINGS+12
                     67     02 FB 002A9       CALLS   #2, OUTPUT_LINES
                        1E  DD 002AC          PUSHL   #30                                         ; 1743
                     68     01 FB 002AE       CALLS   #1, LABEL_AND_HEX_OUTPUT
                        6E  D4 002B1          CLRL    (SP)                                        ; 1746
                        5E  DD 002B3          PUSHL   SP
                     64 A6  9F 002B5          PUSHAB  FAO_STRING                                  ; 1745
                   00A4 C6  9F 002B8          PUSHAB  OUT_ARGLIST
                   0504 C6  9F 002BC          PUSHAB  BYTE30_DESC_TBL
                     30 A6  9F 002C0          PUSHAB  BIT_MSR_0
                     EB A9  9F 002C3          PUSHAB  DRIVE_STS+29
```

```
                        6B              06  FB 002C6          CALLS   #6, TRANSLATE_BITS                       ;
                        0A              50  E9 002C9          BLBC    R0, 15$                                 ;
                                  00A4  C6  DD 002CC          PUSHL   OUT_ARGLIST                             ; 1748
                                    64  A6  DD 002D0          PUSHL   FAO_STRING                              ;
                        67              02  FB 002D3          CALLS   #2, OUTPUT_LINES                        ;
                                        1F  DD 002D6 15$:     PUSHL   #31                                     ; 1754
                        68              01  FB 002D8          CALLS   #1, LABEL_AND_HEX_OUTPUT                ;
                                        6E  D4 002DB          CLRL    (SP)                                    ; 1755
                                        5E  DD 002DD          PUSHL   SP                                      ;
                     7E         EC      A9  9A 002DF          MOVZBL  DRIVE_STS+30, -(SP)                     ;
          00000000v  00                 02  FB 002E3          CALLS   #2, READ_CHANNEL_TIE_DECODE             ;
                                        20  DD 002EA          PUSHL   #32                                     ; 1760
                        68              01  FB 002EC          CALLS   #1, LABEL_AND_HEX_OUTPUT                ;
                        6E              01  D0 002EF          MOVL    #1, (SP)                                ; 1761
                                        5E  DD 002F2          PUSHL   SP                                      ;
                     7E         ED      A9  9A 002F4          MOVZBL  DRIVE_STS+31, -(SP)                     ;
          00000000v  00                 02  FB 002F8          CALLS   #2, READ_CHANNEL_TIE_DECODE             ;
                                        21  DD 002FF          PUSHL   #33                                     ; 1766
                        68              01  FB 00301          CALLS   #1, LABEL_AND_HEX_OUTPUT                ;
                        6E              02  D0 00304          MOVL    #2, (SP)                                ; 1767
                                        5E  DD 00307          PUSHL   SP                                      ;
                     7E         EE      A9  9A 00309          MOVZBL  DRIVE_STS+32, -(SP)                     ;
          00000000v  00                 02  FB 0030D          CALLS   #2, READ_CHANNEL_TIE_DECODE             ;
                                        22  DD 00314          PUSHL   #34                                     ; 1772
                        68              01  FB 00316          CALLS   #1, LABEL_AND_HEX_OUTPUT                ;
                        6E              03  D0 00319          MOVL    #3, (SP)                                ; 1773
                                        5E  DD 0031C          PUSHL   SP                                      ;
                     7E         EF      A9  9A 0031E          MOVZBL  DRIVE_STS+33, -(SP)                     ;
          00000000v  00                 02  FB 00322          CALLS   #2, READ_CHANNEL_TIE_DECODE             ;
                                        23  DD 00329          PUSHL   #35                                     ; 1778
                        68              01  FB 0032B          CALLS   #1, LABEL_AND_HEX_OUTPUT                ;
                              64        A6      0230  CA  9E 0032E          MOVAB   P.ANQ, FAO_STRING               ; 1780
                                        52  D4 00334          CLRL    I                                       ; 1781
                        0E              52  E1 00336 16$:     BBC     I, DRIVE_STS+34, 17$                    ; 1783
                                        66      00AC C642 7E  00338          MOVAQ   BYTE31_DESC_TBL[I], ARGLIST     ; 1786
                                        56  DD 00341          PUSHL   R6                                      ; 1787
                                    64  A6  DD 00343          PUSHL   FAO_STRING                              ;
                                    67      02  FB 00346          CALLS   #2, OUTPUT_LINES                        ;
                        E9              52  F3 00349 17$:     AOBLEQ  #3, I, 16$                              ; 1781
                              64        A6      0250  CA  9E 0034D          MOVAB   P.ANS, FAO_STRING               ; 1791
                66      F0  A9          04      07  EF 00353          EXTZV   #7, #4, DRIVE_STS+34, ARGLIST   ; 1792
                                        56  DD 00359          PUSHL   R6                                      ; 1794
                                    64  A6  DD 0035B          PUSHL   FAO_STRING                              ;
                                    67      02  FB 0035E          CALLS   #2, OUTPUT_LINES                        ;
                                        24  DD 00361          PUSHL   #36                                     ; 179?
                                    68      01  FB 00363          CALLS   #1, LABEL_AND_HEX_OUTPUT                ;
                                    66      0274  CA  9E 00366          MOVAB   P.ANU, ARGLIST                  ; 1801
                                        56  DD 0036B          PUSHL   R6                                      ; 1802
                                    74  A6  DD 0036D          PUSHL   FAO_STRINGS+12                          ;
                                    67      02  FB 00370          CALLS   #2, OUTPUT_LINES                        ;
                                        25  DD 00373          PUSHL   #37                                     ; 1807
                                    68      01  FB 00375          CALLS   #1, LABEL_AND_HEX_OUTPUT                ;
                                        6E  D4 00378          CLRL    (SP)                                    ; 1810
                                        5E  DD 0037A          PUSHL   SP                                      ;
                                    64  A6  9F 0037C          PUSHAB  FAO_STRING                              ; 1809
                                  00A4  C6  9F 0037F          PUSHAB  OUT_ARGLIST                             ;
                                  0544  C6  9F 00383          PUSHAB  BYTE37_DESC_TBL                         ;
```

```
                30 A6 9F 00387          PUSHAB  BIT_MSK_0                                    ;
                F2 A9 9F 0038A          PUSHAB  DRIVE_STS+36                                ;
            6B     06 FB 0038D          CALLS   #6, TRANSLATE_BITS                          ;
            0A     50 E9 00390          BLBC    R0, 18$                                     ;
                00A4 C6 DD 00393          PUSHL   OUT_ARGLIST                               ; 1812
                  64 A6 DD 00397          PUSHL   FAO_STRING
            67     02 FB 0039A          CALLS   #2, OUTPUT_LINES                            ;
                   26 DD 0039D 18$:     PUSHL   #38                                       ; 1818
            68     01 FB 0039F          CALLS   #1, LABEL_AND_HEX_OUTPUT                    ;
            66 0298   CA 9E 003A2          MOVAB   P.ANW, ARGLIST                            ; 1820
                   56 DD 003A7          PUSHL   R6                                        ; 1821
                  74 A6 DD 003A9          PUSHL   FAO_STRINGS+12                            ;
            67     02 FB 003AC          CALLS   #2, OUTPUT_LINES                            ;
                   27 DD 003AF          PUSHL   #39                                       ; 1826
            68     01 FB 003B1          CALLS   #1, LABEL_AND_HEX_OUTPUT                    ;
                   6E D4 003B4          CLRL    (SP)                                      ; 1829
                   5E DD 003B6          PUSHL   SP
                  64 A6 9F 003B8          PUSHAB  FAO_STRING                                ; 1828
                00A4 C6 9F 003BB          PUSHAB  OUT_ARGLIST
                0584 C6 9F 003BF          PUSHAB  BYTE39_DESC_TBL
                  50 A6 9F 003C3          PUSHAB  BIT_MSK_8
                  F4 A9 9F 003C6          PUSHAB  DRIVE_STS+38
            6B     06 FB 003C9          CALLS   #6, TRANSLATE_BITS                          ;
            0A     50 E9 003CC          BLBC    R0, 19$                                     ;
                00A4 C6 DD 003CF          PUSHL   OUT_ARGLIST                               ; 1831
                  64 A6 DD 003D3          PUSHL   FAO_STRING
            67     02 FB 003D6          CALLS   #2, OUTPUT_LINES                            ;
                   28 DD 003D9 19$:     PUSHL   #40                                       ; 1836
            68     01 FB 003DB          CALLS   #1, LABEL_AND_HEX_OUTPUT                    ;
                   6E D4 003DE          CLRL    (SP)                                      ; 1839
                   5E DD 003E0          PUSHL   SP
                  64 A6 9F 003E2          PUSHAB  FAO_STRING                                ; 1838
                00A4 C6 9F 003E5          PUSHAB  OUT_ARGLIST
                05BC C6 9F 003E9          PUSHAB  BYTE40_DESC_TBL
                  30 A6 9F 003ED          PUSHAB  BIT_MSK_0
                  F5 A9 9F 003F0          PUSHAB  DRIVE_STS+39
            6B     06 FB 003F3          CALLS   #6, TRANSLATE_BITS                          ;
            0A     50 E9 003F6          BLBC    R0, 20$                                     ;
                00A4 C6 DD 003F9          PUSHL   OUT_ARGLIST                               ; 1841
                  64 A6 DD 003FD          PUSHL   FAO_STRING
            67     02 FB 00400          CALLS   #2, OUTPUT_LINES                            ;
                   29 DD 00403 20$:     PUSHL   #41                                       ; 1846
            68     01 FB 00405          CALLS   #1, LABEL_AND_HEX_OUTPUT                    ;
            66 02DC   C6 D0 00408          MOVL    TU_STRING, ARGLIST                        ; 1848
04  A6      F6 A9 02     00 EF 0040D          EXTZV   #0, #2, DRIVE_STS+40, ARGLIST+4           ; 1849
                   56 DD 00414          PUSHL   R6                                        ; 1851
                  70 A6 DD 00416          PUSHL   FAO_STRINGS+8                             ;
            67     02 FB 00419          CALLS   #2, OUTPUT_LINES                            ;
                   6E D4 0041C          CLRL    (SP)                                      ; 1854
                   5E DD 0041E          PUSHL   SP
                  64 A6 9F 00420          PUSHAB  FAO_STRING                                ; 1853
                00A4 C6 9F 00423          PUSHAB  OUT_ARGLIST
                05FC C6 9F 00427          PUSHAB  BYTE41_DESC_TBL
                  3C A6 9F 0042B          PUSHAB  BIT_MSK_3
                  F6 A9 9F 0042E          PUSHAB  DRIVE_STS+40
            6B     06 FB 00431          CALLS   #6, TRANSLATE_BITS                          ;
            0A     50 E9 00434          BLBC    R0, 21$                                     ;
```

```
                                                00A4   C6   DD 00437          PUSHL   OUT_ARGLIST                                    ; 1856
                                                  64   A6   DD 0043B          PUSHL   FAO_STRING
                                                 67    02   FB 0043E          CALLS   #2, OUTPUT_LINES                              ;
                                                       2A   DD 00441 21$:     PUSHL   #42                                           ; 1861
                                                 68    01   FB 00443          CALLS   #1, LABEL_AND_HEX_OUTPUT                      ;
                                                 66  02DC   C6   D0 00446          MOVL    TU_STRING, ARGLIST                    ; 1863
    04   A6        F7   A9                      02         00   EF 00448          EXTZV   #0, #2, DRIVE_STS+41, ARGLIST+4       ; 1864
                                                       56   DD 00452          PUSHL   R6                                            ; 1866
                                                   70  A6   DD 00454          PUSHL   FAO_STRINGS+8                                 ;
                                                 67    02   FB 00457          CALLS   #2, OUTPUT_LINES                              ;
                                                       6E   D4 0045A          CLRL    (SP)                                          ; 1869
                                                       5E   DD 0045C          PUSHL   SP                                            ;
                                                   64  A6   9F 0045E          PUSHAB  FAO_STRING                                    ; 1868
                                                 00A4  C6   9F 00461          PUSHAB  OUT_ARGLIST                                   ;
                                                 062C  C6   9F 00465          PUSHAB  BYTE42_DESC_TBL                               ;
                                                   58  A6   9F 00469          PUSHAB  BIT_MSK_10                                    ;
                                                   F7  A9   9F 0046C          PUSHAB  DRIVE_STS+41                                  ;
                                                 6B    06   FB 0046F          CALLS   #6, TRANSLATE_BITS                            ;
                                                 0A    50   E9 00472          BLBC    R0, 22$                                       ;
                                                 00A4  C6   DD 00475          PUSHL   OUT_ARGLIST                                   ; 1871
                                                   64  A6   DD 00479          PUSHL   FAO_STRING                                    ;
                                                 67    02   FB 0047C          CALLS   #2, OUTPUT_LINES                              ;
                                                       2B   DD 0047F 22$:     PUSHL   #43                                           ; 1877
                                                 68    01   FB 00481          CALLS   #1, LABEL_AND_HEX_OUTPUT                      ;
                                                 66  02C0   CA   9E 00484          MOVAB   P.ANY, ARGLIST                        ; 1879
                                                       56   DD 00489          PUSHL   R6                                            ; 1880
                                                   74  A6   DD 0048B          PUSHL   FAO_STRINGS+12                                ;
                                                 67    02   FB 0048E          CALLS   #2, OUTPUT_LINES                              ;
                                                       2D   DD 00491          PUSHL   #45                                           ; 1885
                                                       2C   DD 00493          PUSHL   #44                                           ;
                                                 68    02   FB 00495          CALLS   #2, LABEL_AND_HEX_OUTPUT                      ;
                                              64   A6  02E0   CA   9E 00498          MOVAB   P.AOA, FAO_STRING                     ; 1887
                                                 66    F9   A9   3C 0049E          MOVZWL  BYTE_COUNT, ARGLIST                   ; 1888
                                                       56   DD 004A2          PUSHL   R6                                            ; 1890
                                                   64  A6   DD 004A4          PUSHL   FAO_STRING                                    ;
                                                 67    02   FB 004A7          CALLS   #2, OUTPUT_LINES                              ;
                                                       2F   DD 004AA          PUSHL   #47                                           ; 1895
                                                       2E   DD 004AC          PUSHL   #46                                           ;
                                                 68    02   FB 004AE          CALLS   #2, LABEL_AND_HEX_OUTPUT                      ;
                                              64   A6  0304   CA   9E 004B1          MOVAB   P.AOC, FAO_STRING                     ; 1897
                                                 66    FB   A9   3C 004B7          MOVZWL  PAD_COUNT, ARGLIST                    ; 1898
                                                       56   DD 004BB          PUSHL   R6                                            ; 1900
                                                   64  A6   DD 004BD          PUSHL   FAO_STRING                                    ;
                                                 67    02   FB 004C0          CALLS   #2, OUTPUT_LINES                              ;
                                                       31   DD 004C3          PUSHL   #49                                           ; 190[illegible]
                                                       30   DD 004C5          PUSHL   #48                                           ;
                                                 68    02   FB 004C7          CALLS   #2, LABEL_AND_HEX_OUTPUT                      ;
                                              64   A6    74   A6   D0 004CA          MOVL    FAO_STRINGS+12, FAO_STRING            ; 1907
                                                 50    FD   A9   3C 004CF          MOVZWL  ERR_CODE_CNT, R0                      ; 1908
                                  05 FFFFFFFB    8F         50   CF 004D3          CASEL   R0, #-5, #5                                   ;
    0039          004C                 0053                 005A      004DB 23$:     .WORD   29$-23$,-                             ;
                                       0013                 001A      004E3                  28$-23$,-                              ;
                                                                                              27$-23$,-                              ;
                                                                                              26$-23$,-                              ;
                                                                                              25$-23$,-                              ;
                                                                                              24$-23$                                ;
                                                 66  04B8   CA   9E 004E7          MOVAB   P.APC, ARGLIST                        ; 1934
```

```
                        4C  11 004EC             BRB     30$                                                     ;
                66      0324    CA  9E 004EE 24$:       MOVAB   P.AOE, ARGLIST                                  ; 1910
                        45  11 004F3             BRB     30$                                                     ;
                66      0344    CA  9E 004F5 25$:       MOVAB   P.AOG, ARGLIST                                  ; 1913
             04 A6      0360    CA  9E 004FA            MOVAB   P.AOI, ARGLIST+4                                ; 1914
             08 A6      0380    CA  9E 00500            MOVAB   P.AOK, ARGLIST+8                                ; 1915
             0C A6      03A8    CA  9E 00506            MOVAB   P.AOM, ARGLIST+12                               ; 1916
             64 A6      03E0    CA  9E 0050C            MOVAB   P.AOO, FAO_STRING                               ; 1918
                        26  11 00512             BRB     30$                                                     ; 1908
                66      0400    CA  9E 00514 26$:       MOVAB   P.AOQ, ARGLIST                                  ; 1922
             04 A6      0418    CA  9E 00519            MOVAB   P.AOS, ARGLIST+4                                ; 1923
             64 A6      0438    CA  9E 0051F            MOVAB   P.AOU, FAO_STRING                               ; 1925
                        13  11 00525             BRB     30$                                                     ; 1908
                66      0450    CA  9E 00527 27$:       MOVAB   P.AOW, ARGLIST                                  ; 1928
                        0C  11 00532             BRB     30$                                                     ;
                66      0478    CA  9E 00534 28$:       MOVAB   P.AOY, ARGLIST                                  ; 1930
                        05  11 00539             BRB     30$                                                     ;
                66      049C    CA  9E 0053B 29$:       MOVAB   P.APA, ARGLIST                                  ; 1932
                        56  DD 00540 30$:       PUSHL   R6                                                      ; 1937
                   64   A6  DD 00542            PUSHL   FAO_STRING                                              ;
                67      02  FB 00545            CALLS   #2, OUTPUT_LINES                                        ;
                        32  DD 00548            PUSHL   #50                                                     ; 1942
                68      01  FB 0054A            CALLS   #1, LABEL_AND_HEX_OUTPUT                                ;
                        6E  D4 0054D            CLRL    (SP)                                                    ; 1945
                        5E  DD 0054F            PUSHL   SP                                                      ;
                   64   A6  9F 00551            PUSHAB  FAO_STRING                                              ; 1944
                 00A4   C6  9F 00554            PUSHAB  OUT_ARGLIST                                             ;
                 064C   C6  9F 00558            PUSHAB  BYTE50_DESC_TBL                                         ;
                   48   A6  9F 0055C            PUSHAB  BIT_MSR_6                                               ;
                   FF   A9  9F 0055F            PUSHAB  DRIVE_STS+49                                            ;
                6B      06  FB 00562            CALLS   #6, TRANSLATE_BITS                                      ;
                0A      50  E9 00565            BLBC    R0, 31$                                                 ;
                 00A4   C6  DD 00568            PUSHL   OUT_ARGLIST                                             ; 1947
                   64   A6  DD 0056C            PUSHL   FAO_STRING                                              ;
                67      02  FB 0056F            CALLS   #2, OUTPUT_LINES                                        ;
                        33  DD 00572 31$:       PUSHL   #51                                                     ; 1952
                68      01  FB 00574            CALLS   #1, LABEL_AND_HEX_OUTPUT                                ;
                        6E  D4 00577            CLRL    (SP)                                                    ; 1955
                        5E  DD 00579            PUSHL   SP                                                      ;
                   64   A6  9F 0057B            PUSHAB  FAO_STRING                                              ; 1954
                 00A4   C6  9F 0057E            PUSHAB  OUT_ARGLIST                                             ;
                 066C   C6  9F 00582            PUSHAB  BYTE51_DESC_TBL                                         ;
                   60   A6  9F 00586            PUSHAB  BIT_MSR_12                                              ;
                        59  DD 00589            PUSHL   R9                                                      ;
                6B      06  FB 0058B            CALLS   #6, TRANSLATE_BITS                                      ;
                0A      50  E9 0058E            BLBC    R0, 32$                                                 ;
                 00A4   C6  DD 00591            PUSHL   OUT_ARGLIST                                             ; 1957
                   64   A6  DD 00595            PUSHL   FAO_STRING                                              ;
                67      02  FB 00598            CALLS   #2, OUTPUT_LINES                                        ;
             0D 69      02  E0 0059B 32$:       BBS     #2, DRIVE_STS+50, 33$                                   ; 1959
                66      04DC    CA  9E 005A0            MOVAB   P.APE, ARGLIST                                  ; 1962
                        56  DD 005A5            PUSHL   R6                                                      ; 1963
                   74   A6  DD 005A7            PUSHL   FAO_STRINGS+12                                          ;
                67      02  FB 005AA            CALLS   #2, OUTPUT_LINES                                        ;
                        6E  D4 005AD 33$:       CLRL    (SP)                                                    ; 1967
                        5E  DD 005AF            PUSHL   SP                                                      ;
                   64   A6  9F 005B1            PUSHAB  FAO_STRING                                              ; 1966
```

```
                00A4   C6  9F 005AD          PUSHAB  OUT_ARGLIST                                    :
                067C   C6  9F 005B1          PUSHAB  BYTES1_2_DESC_TBL                              :
                  4C   A6  9F 005B5          PUSHAB  BIT_MSK_7                                      :
                       59  DD 005B8          PUSHL   R9                                             :
            6B         06  FB 005BA          CALLS   #6, TRANSLATE_BITS                             :
            0A         50  E9 005BD          BLBC    R0, 34$                                        :
                00A4   C6  DD 005C0          PUSHL   OUT_ARGLIST                                    : 1969
                  64   A6  DD 005C4          PUSHL   FAO_STRING                                     :
            67         02  FB 005C7          CALLS   #2, OUTPUT_LINES                               :
            6E         01  DD 005CA 34$:      MOVL    #1, (SP)                                       : 1972
                       5E  DD 005CD          PUSHL   SP                                             :
                  64   A6  9F 005CF          PUSHAB  FAO_STRING                                     : 1971
                00A4   C6  9F 005D2          PUSHAB  OUT_ARGLIST                                    :
                0694   C6  9F 005D6          PUSHAB  BYTES1_3_DESC_TBL                              :
                  3C   A6  9F 005DA          PUSHAB  BIT_MSK_3                                      :
                       59  DD 005DD          PUSHL   R9                                             :
            6B         06  FB 005DF          CALLS   #6, TRANSLATE_BITS                             :
            0A         50  E9 005E2          BLBC    R0, 35$                                        :
                00A4   C6  DD 005E5          PUSHL   OUT_ARGLIST                                    : 1974
                  64   A6  DD 005E9          PUSHL   FAO_STRING                                     :
            67         02  FB 005EC          CALLS   #2, OUTPUT_LINES                               :
                       34  DD 005EF 35$:      PUSHL   #52                                            : 1980
            68         01  FB 005F1          CALLS   #1, LABEL_AND_HEX_OUTPUT                       :
                       6E  D4 005F4          CLRL    (SP)                                           : 1983
                       5E  DD 005F6          PUSHL   SP                                             :
                  64   A6  9F 005F8          PUSHAB  FAO_STRING                                     : 1982
                00A4   C6  9F 005FB          PUSHAB  OUT_ARGLIST                                    :
                06A4   C6  9F 005FF          PUSHAB  BYTES2_DESC_TBL                                :
                  30   A6  9F 00603          PUSHAB  BIT_MSK_0                                      :
                  01   A9  9F 00606          PUSHAB  DRIVE_STS+51                                   :
            6B         06  FB 00609          CALLS   #6, TRANSLATE_BITS                             :
            0A         50  E9 0060C          BLBC    R0, 36$                                        :
                00A4   C6  DD 0060F          PUSHL   OUT_ARGLIST                                    : 1985
                  64   A6  DD 00613          PUSHL   FAO_STRING                                     :
            67         02  FB 00616          CALLS   #2, OUTPUT_LINES                               :
                       35  DD 00619 36$:      PUSHL   #53                                            : 1990
            68         01  FB 0061B          CALLS   #1, LABEL_AND_HEX_OUTPUT                       :
                       6E  D4 0061E          CLRL    (SP)                                           : 1993
                       5E  DD 00620          PUSHL   SP                                             :
                  64   A6  9F 00622          PUSHAB  FAO_STRING                                     : 1992
                00A4   C6  9F 00625          PUSHAB  OUT_ARGLIST                                    :
                06E4   C6  9F 00629          PUSHAB  BYTES3_DESC_TBL                                :
                  30   A6  9F 0062D          PUSHAB  BIT_MSK_0                                      :
                  02   A9  9F 00630          PUSHAB  DRIVE_STS+52                                   :
            6B         06  FB 00633          CALLS   #6, TRANSLATE_BITS                             :
            0A         50  E9 00636          BLBC    R0, 37$                                        :
                00A4   C6  DD 00639          PUSHL   OUT_ARGLIST                                    : 1995
                  64   A6  DD 0063D          PUSHL   FAO_STRING                                     :
            67         02  FB 00640          CALLS   #2, OUTPUT_LINES                               :
                       36  DD 00643 37$:      PUSHL   #54                                            : 2001
            68         01  FB 00645          CALLS   #1, LABEL_AND_HEX_OUTPUT                       :
      04                55  CF 00648          CASEL   R5, #3, #2                                     : 2003
0020  0019         000A         0012  0064C 38$:      .WORD   40$-38$,-                              :
                           0027  00654                  39$-38$,-                                  :
                                                       41$-38$,-                                  :
                                                       42$-38$,-                                  :
                                                       43$-38$                                    :
```

```
                04  A6      056C    CA  9E 00656 39$:    MOVAB   P.APO, ARGLIST+4                    ; 2010
                                    1A  11 0065C         BRB     44$
                    66      04F8    CA  9E 0065E 40$:    MOVAB   P.APG, ARGLIST                     ; 2005
                                    13  11 00663         BRB     44$
                    66      0514    CA  9E 00665 41$:    MOVAB   P.API, ARGLIST                     ; 2006
                                    0C  11 0066A         BRB     44$
                    66      0530    CA  9E 0066C 42$:    MOVAB   P.APK, ARGLIST                     ; 2007
                                    05  11 00671         BRB     44$
                    66      054C    CA  9E 00673 43$:    MOVAB   P.APM, ARGLIST                     ; 2008
                04  A6      0588    CA  9E 00678 44$:    MOVAB   P.APQ, ARGLIST+4                   ; 2013
                                  04    A6  9F 0067E       PUSHAB  ARGLIST+4                          ; 2016
                    50          74    A6  D0 00681       MOVL    FAO_STRINGS+12, R0
                                  50    DD 00685         PUSHL   R0
                            0041        8F  BB 00687       PUSHR   #^M<R0,R6>                         ; 2015
                    67                  04  FB 0068B       CALLS   #4, OUTPUT_LINES
                                        6E  D4 0068E       CLRL    (SP)                               ; 2020
                                        5E  DD 00690       PUSHL   SP
                                    64      A6  9F 00692   PUSHAB  FAO_STRING                         ; 2019
                                  00A4      C6  9F 00695   PUSHAB  OUT_ARGLIST
                                  0724      C6  9F 00699   PUSHAB  BYTE54_DESC_TBL
                                    54      A6  9F 0069D   PUSHAB  BIT_MSR_9
                                    03      A9  9F 006A0   PUSHAB  DRIVE_STS+53
                    6B                      06  FB 006A3   CALLS   #6, TRANSLATE_BITS
                    0A                      50  E9 006A6   BLBC    R0, 45$
                                  00A4      C6  DD 006A9   PUSHL   OUT_ARGLIST                        ; 2022
                                    64      A6  DD 006AD   PUSHL   FAO_STRING
                    67                      02  FB 006B0   CALLS   #2, OUTPUT_LINES
                                            38  DD 006B3 45$:  PUSHL   #56                          ; 2028
                                            37  DD 006B5       PUSHL   #55
                        69                  02  FB 006B7   CALLS   #2, LABEL_AND_HEX_OUTPUT
                        66          05A4    CA  9E 006BA   MOVAB   P.APS, ARGLIST                    ; 2030
                  02E0  C6                  54  90 006BF   MOVB    R4, SERIAL_NUMBER                ; 2032
                  02E1  C6                  53  90 006C4   MOVB    R3, SERIAL_NUMBER+1              ; 2033
                                            04  DD 006C9   PUSHL   #4                               ; 2034
                        7E          02E0    C6  3C 006CB   MOVZWL  SERIAL_NUMBER, -(SP)
                00000000V   00              02  FB 006D0   CALLS   #2, CONVERT_BCD_NUMBER
                      04    A6              50  D0 006D7   MOVL    R0, ARGLIST+4
                                            56  DD 006DB   PUSHL   R6                              ; 2036
                                    0098    C6  DD 006DD   PUSHL   FAO_STRINGS+48
                        67                  02  FB 006E1   CALLS   #2, OUTPUT_LINES
                                            39  DD 006E4   PUSHL   #57                             ; 2041
                        68                  01  FB 006E6   CALLS   #1, LABEL_AND_HEX_OUTPUT
                        66          05C4    CA  9E 006E9   MOVAB   P.APU, ARGLIST                    ; 2043
04  A6      06  A9                  02      00  EF 006EE   EXTZV   #0, #2, DRIVE_STS+56, ARGLIST+4  ; 2044
                                            56  DD 006F5   PUSHL   R6                              ; 2045
                                      70    A6  DD 006F7   PUSHL   FAO_STRINGS+8
                        67                  02  FB 006FA   CALLS   #2, OUTPUT_LINES
                                            6E  D4 006FD   CLRL    (SP)                            ; 2049
                                            5E  DD 006FF   PUSHL   SP
                                    64      A6  9F 00701   PUSHAB  FAO_STRING                       ; 2048
                                  00A4      C6  9F 00704   PUSHAB  OUT_ARGLIST
                                  073C      C6  9F 00708   PUSHAB  BYTE57_DESC_TBL
                                    38      A6  9F 0070C   PUSHAB  BIT_MSR_2
                                    06      A9  9F 0070F   PUSHAB  DRIVE_STS+56
                        6B                      06  FB 00712   CALLS   #6, TRANSLATE_BITS
                        0A                      50  E9 00715   BLBC    R0, 46$
                                  00A4      C6  DD 00718   PUSHL   OUT_ARGLIST                       ; 2051
```

```
                                64    A6  DD 0071C          PUSHL   FAO_STRING                                    ;
                        67          02  FB 0071F          CALLS   #2, OUTPUT_LINES                               ;
                                    3B  DD 00722 468$:     PUSHL   #59                                            ; 2056
                                    3A  DD 00724           PUSHL   #58                                            ;
                        68          02  FB 00726          CALLS   #2, LABEL_AND_HEX_OUTPUT                       ;
                                        04 00729          RET                                                    ; 2059

; Routine Size:  1834 bytes,    Routine Base:  $CODE + 05B9

; 1651          2060  1
```

```
;  1653          2061  1 Routine READ_CHANNEL_TIE_DECODE (channel_tie_byte,flag) : NOVALUE =
;  1654          2062  2 Begin
;  1655          2063  2 !++
;  1656          2064  2 !
;  1657          2065  2 ! Functional Description:
;  1658          2066  2 !
;  1659          2067  2 !       This routine decodes and outputs the read channel tie byte. It is
;  1660          2068  2 !       called from the TA78_drive_sts_decode routine.
;  1661          2069  2 !
;  1662          2070  2 ! Calling Sequence:
;  1663          2071  2 !
;  1664          2072  2 !       READ_CHANNEL_TIE_DECODE (channel_tie_byte,flag) ;
;  1665          2073  2 !
;  1666          2074  2 ! Input Parameters:
;  1667          2075  2 !
;  1668          2076  2 !       Channel_tie_byte = contents of the read channel tie byte.
;  1669          2077  2 !       Flag = address of bus selection indicator.
;  1670          2078  2 !
;  1671          2079  2 ! Output Parameters:
;  1672          2080  2 !
;  1673          2081  2 !       None.
;  1674          2082  2 !
;  1675          2083  2 !--
;  1676          2084  2 Own
;  1677          2085  2     Unknown_desc:       Initial (%ASCID '?') ;
;  1678          2086  2
;  1679          2087  2 Fao_string = %ASCID '!39< !>CHANNEL !AS TIE BUS !AS' ;
;  1680          2088  2
;  1681          2089  2 Incr I from 0 to 7 do
;  1682          2090  3     Begin
;  1683          2091  3     If .channel_tie_byte<.I>
;  1684          2092  3     Then
;  1685          2093  4         Begin
;  1686          2094  4         If .I LEQ 3
;  1687          2095  4         Then
;  1688          2096  5             Begin
;  1689          2097  5             Arglist[1] = byte31_desc_tbl[.I,0,0,0,0] ;
;  1690          2098  5
;  1691          2099  5             Case ..flag from 0 to 3 of
;  1692          2100  5                 Set
;  1693          2101  5                 [0]: Arglist[0] = byte31_desc_tbl[0,0,0,0,0] ;
;  1694          2102  5                 [1]: Arglist[0] = byte31_desc_tbl[2,0,0,0,0] ;
;  1695          2103  5                 [2]: Arglist[0] = byte31_desc_tbl[4,0,0,0,0] ;
;  1696          2104  5                 [3]: Arglist[0] = byte31_desc_tbl[6,0,0,0,0] ;
;  1697          2105  5
;  1698          2106  5                 [OUTRANGE]: Arglist[0] = .unknown_desc ;
;  1699          2107  5                 Tes ;
;  1700          2108  5             End
;  1701          2109  4         Else
;  1702          2110  5             Begin
;  1703          2111  5             Arglist[1] = byte31_desc_tbl[(.I-4),0,0,0,0] ;
;  1704          2112  5
;  1705          2113  5             Case ..flag from 0 to 3 of
;  1706          2114  5                 Set
;  1707          2115  5                 [0]: Arglist[0] = byte31_desc_tbl[1,0,0,0,0] ;
;  1708          2116  5                 [1]: Arglist[0] = byte31_desc_tbl[3,0,0,0,0] ;
;  1709          2117  5                 [2]: Arglist[0] = byte31_desc_tbl[5,0,0,0,0] ;
```

```
;  1710          2118  5                  [3]: Arglist[0] = byte31_desc_tbl[7,0,0,0,0] ;
;  1711          2119  5
;  1712          2120  5                  [OUTRANGE]: Arglist[0] = .unknown_desc ;
;  1713          2121  5                  Tes ;
;  1714          2122  4              End ;
;  1715          2123  3          End ;
;  1716          2124  2      End ;
;  1717          2125  2
;  1718          2126  2  If .channel_tie_byte NEQ 0
;  1719          2127  2  Then
;  1720          2128  2      OUTPUT_LINES (  .fao_string, arglist[0]  ) ;
;  1721          2129  2
;  1722          2130  2  Return ;
;  1723          2131  1  End ;   ! Routine


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

                                                    00  00  00  3F  01E08 P.APX:  .ASCII  \?\<0><0><0>                                  ;
                                                      010E0001  01E0C P.APW:  .LONG   17694721                                         ;
                                                      00000000' 01E10         .ADDRESS P.APX                                           ;
20  4C  45  4E  4E  41  48  43  3E  21  20  3C  39  33  21  01E14 P.APZ:  .ASCII  \!39< !>CHANNEL !AS TIE BUS !AS\<0><0>                ;
53  41  21  20  53  55  42  20  45  49  54  20  53  41  21  01E23                                                                       ;
                                                        00  00  01E32                                                                  ;
                                                      010E001E  01E34 P.APY:  .LONG   17694750                                         ;
                                                      00000000' 01E38         .ADDRESS P.APZ                                           ;

                                                                .PSECT  $OWN$,NOEXE,  PIC,2

                                                00000000' 00764 UNKNOWN_DESC:
                                                                .ADDRESS P.APW                                                         ;


                                                                .PSECT  $CODE,NOWRT,  PIC,2

                                                    0004 00000 READ_CHANNEL_TIE_DECODE:
                                                                .WORD   Save R2                                                        ; 2061
                                  52 00000000'  00  9E 00002    MOVAB   ARGLIST, R2                                                     ;
                                64  A2 00000000'  00  9E 00009    MOVAB   P.APY, FAO_STRING                                            ; 2087
                                                  50  D4 00011    CLRL    I                                                            ; 2089
                              6C  04  AC          50  E1 00013 1$:  BBC     I, CHANNEL_TIE_BYTE, 14$                                     ; 2091
                                  03              50  D1 00018    CMPL    I, #3                                                        ; 2094
                                                  32  14 0001B    BGTR    7$                                                           ;
                                04  A2      00AC C240  7E 0001D    MOVAQ   BYTE31_DESC_TBL[I], ARGLIST+4                                ; 2097
                            03      00        08  BC  CF 00024    CASEL   @FLAG, #0, #3                                                ; 2099
                001F      0018          0011        000A  00029 2$:  .WORD   3$-2$,-                                                    ;
                                                                             4$-2$,-                                                   ;
                                                                             5$-2$,-                                                   ;
                                                                             6$-2$                                                     ;
                                                  30  11 00031    BRB     9$                                                           ; 2106
                                  62      00AC    C2  9E 00033 3$:  MOVAB   BYTE31_DESC_TBL, ARGLIST                                     ; 2101
                                                  4A  11 00038    BRB     14$                                                          ;
                                  62      00BC    C2  9E 0003A 4$:  MOVAB   BYTE31_DESC_TBL+16, ARGLIST                                  ; 2102
                                                  43  11 0003F    BRB     14$                                                          ;
                                  62      00CC    C2  9E 00041 5$:  MOVAB   BYTE31_DESC_TBL+32, ARGLIST                                  ; 2103
```

```
                                        3C  11 00046       BRB     14$                                   ;
                       62        00DC   C2  9E 00048 6$:   MOVAB   BYTE31_DESC_TBL+48, ARGLIST           ; 2104
                                        35  11 0004D       BRB     14$                                   ; 2094
                  04   A2        008C C240  7E 0004F 7$:   MOVAQ   BYTE31_DESC_TBL-32[I], ARGLIST+4      ; 2111
            03         00          08   BC  CF 00056       CASEL   @FLAG, #0, #3                         ; 2113
  0024      001D       0016          000F      0005B 8$:   .WORD   10$-8$,-                              ;
                                                                   11$-8$,-                              ;
                                                                   12$-8$,-                              ;
                                                                   13$-8$                                ;
                       62        0764   C2  D0 00063 9$:   MOVL    UNKNOWN_DESC, ARGLIST                 ; 2120
                                        1A  11 00068       BRB     14$                                   ;
                       62        00B4   C2  9E 0006A 10$:  MOVAB   BYTE31_DESC_TBL+8, ARGLIST            ; 2115
                                        13  11 0006F       BRB     14$                                   ;
                       62        00C4   C2  9E 00071 11$:  MOVAB   BYTE31_DESC_TBL+24, ARGLIST           ; 2116
                                        0C  11 00076       BRB     14$                                   ;
                       62        00D4   C2  9E 00078 12$:  MOVAB   BYTE31_DESC_TBL+40, ARGLIST           ; 2117
                                        05  11 0007D       BRB     14$                                   ;
                       62        00E4   C2  9E 0007F 13$:  MOVAB   BYTE31_DESC_TBL+56, ARGLIST           ; 2118
            8B         50               07  F3 00084 14$:  AOBLEQ  #7, I, 1$                             ; 2089
                            04          AC  D5 00088       TSTL    CHANNEL_TIE_BYTE                      ; 2126
                                        0C  13 0008B       BEQL    15$                                   ;
                                        52  DD 0008D       PUSHL   R2                                    ; 2128
                            64          A2  DD 0008F       PUSHL   FAO_STRING                            ;
              00000000G  00             02  FB 00092       CALLS   #2, OUTPUT_LINES                      ;
                                        04     00099 15$:  RET                                           ; 2131

; Routine Size:  154 bytes,    Routine Base:  $CODE + 0CE3


; 1724         2132  1
```

```
;  1726          2133  1 Routine LABEL_AND_HEX_OUTPUT : NOVALUE =
;  1727          2134  2 Begin
;  1728          2135  2
;  1729          2136  2 !++
;  1730          2137  2 !
;  1731          2138  2 ! Functional Description:
;  1732          2139  2 !
;  1733          2140  2 !       This routine outputs the data in hex and an associated label.
;  1734          2141  2 !
;  1735          2142  2 ! Calling Sequence:
;  1736          2143  2 !
;  1737          2144  2 !       LABEL_AND_HEX_OUTPUT (byte_number,byte_number,...) ;
;  1738          2145  2 !
;  1739          2146  2 ! Input Parameters:
;  1740          2147  2 !
;  1741          2148  2 !       Byte_number = actual byte number.
;  1742          2149  2 !
;  1743          2150  2 !       This routine will calculate the number of parameters that were
;  1744          2151  2 !       passed to it.
;  1745          2152  2 !
;  1746          2153  2 ! Output Parameters:
;  1747          2154  2 !
;  1748          2155  2 !       None.
;  1749          2156  2 !
;  1750          2157  2 !--
;  1751          2158  2 Bind
;  1752          2159  2     Hex_data = emb[82,0,0,0] : VECTOR [,byte] ;
;  1753          2160  2
;  1754          2161  2 Builtin
;  1755          2162  2     Actualcount,
;  1756          2163  2     Actualparameter ;
;  1757          2164  2
;  1758          2165  2 Local
;  1759          2166  2     Byte_number,
;  1760          2167  2     Digit_size ;
;  1761          2168  2
;  1762          2169  2 Literal
;  1763          2170  2     Str_size = 5 ;
;  1764          2171  2
;  1765          2172  2 Digit_size = 0 ;
;  1766          2173  2
;  1767          2174  2 Incr I from 1 to ACTUALCOUNT() do
;  1768          2175  3     Begin
;  1769          2176  3     Byte_number = ACTUALPARAMETER (.I) ;
;  1770          2177  3
;  1771          2178  3     If .byte_number GEQ 10
;  1772          2179  3     Then
;  1773          2180  3         Digit_size = 1 ;
;  1774          2181  3
;  1775          2182  3     Arglist[0] = CSTRING ('BYTE ') ;
;  1776          2183  3     Arglist[1] = .byte_number ;
;  1777          2184  3     Arglist[2] = (16 - (str_size +.digit_size)) + 4 ;
;  1778          2185  3     Arglist[3] = .hex_data[(.byte_number-1)] ;
;  1779          2186  3
;  1780          2187  4     If NOT (.main_hdr)
;  1781          2188  3     Then
;  1782          2189  3         Fao_string = .fao_strings[13]
```

```
; 1783        2190  3       Else
; 1784        2191  4           Begin
; 1785        2192  4           Fao_string = .fao_strings[9] ;
; 1786        2193  4           Main_hdr = false ;
; 1787        2194  3           End ;
; 1788        2195  3
; 1789        2196  3       OUTPUT_LINES (  .fao_string, arglist[0]  ) ;
; 1790        2197  2       End ;
; 1791        2198  2
; 1792        2199  2 Return ;
; 1793        2200  1 End ;    ! Routine


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

                    20  45  54  59  42  05  01E3C P.AQA:    .ASCII  <5>\BYTE \                                 ;


                                                            .PSECT  $CODE,NOWRT,  PIC,2

                                       007C 00000 LABEL_AND_HEX_OUTPUT:
                                                            .WORD   Save R2,R3,R4,R5,R6                         ; 2133
                      56 00000000'  00  9E 00002            MOVAB   FAO_STRING, R6                              ;
                                    54  D4 00009            CLRL    DIGIT_SIZE                                  ; 2172
                      55            6C  9A 0000B            MOVZBL  (AP), R5                                    ; 2174
                                    52  D4 0000E            CLRL    I                                           ;
                                    43  11 00010            BRB     5$                                          ;
                      53          6C42  D0 00012 1$:        MOVL    (AP)[I], BYTE_NUMBER                        ; 2176
                      0A            53  D1 00016            CMPL    BYTE_NUMBER, #10                            ; 2178
                                    03  19 00019            BLSS    2$                                          ;
                      54            01  D0 0001B            MOVL    #1, DIGIT_SIZE                              ; 2180
                 9C   A6 00000000'  00  9E 0001E 2$:        MOVAB   P.AQA, ARGLIST                              ; 2182
                 A0   A6            53  D0 00026            MOVL    BYTE_NUMBER, ARGLIST+4                      ; 2183
      A4    A6        0F            54  C3 0002A            SUBL3   DIGIT_SIZE, #15, ARGLIST+8                  ; 2184
                 A8   A6 00000000G0043  9A 0002F            MOVZBL  HEX_DATA-1[BYTE_NUMBER], ARGLIST+12         ; 2185
                      06        3C  A6  E8 00038            BLBS    MAIN_HDR, 3$                                ; 2187
                      66        38  A6  D0 0003C            MOVL    FAO_STRINGS+52, FAO_STRING                  ; 2189
                                    07  11 00040            BRB     4$                                          ;
                      66        28  A6  D0 00042 3$:        MOVL    FAO_STRINGS+36, FAO_STRING                  ; 2192
                                3C  A6  94 00046            CLRB    MAIN_HDR                                    ; 2193
                                9C  A6  9F 00049 4$:        PUSHAB  ARGLIST                                     ; 2196
                                    66  DD 0004C            PUSHL   FAO_STRING                                  ;
              00000000G  00         02  FB 0004E            CALLS   #2, OUTPUT_LINES                            ;
            B9           52         55  F3 00055 5$:        AOBLEQ  R5, I, 1$                                   ; 2174
                                    04  00059               RET                                                 ; 2200

; Routine Size:  90 bytes,    Routine Base:  $CODE + 0D7D


; 1794        2201  1
; 1795        2202  1
```

```
; 1797          2203  1 Routine DIAGNOSTIC_STS_BYTE_DECODE (which_byte,sts_codes_tbl,number_of_codes,
; 1798          2204  1                                                   sts_codes_desc_tbl) : NOVALUE =
; 1799          2205  2 Begin
; 1800          2206  2 !++
; 1801          2207  2 !
; 1802          2208  2 ! Functional Description:
; 1803          2209  2 !
; 1804          2210  2 !       This routine decodes and outputs the diagnostic status byte. It is
; 1805          2211  2 !       called from the TA78_drive_sts_decode routine.
; 1806          2212  2 !
; 1807          2213  2 ! Calling Sequence:
; 1808          2214  2 !
; 1809          2215  2 !       DIAGNOSTIC_STS_BYTE_DECODE (which_byte,sts_codes_tbl,number_of_codes,
; 1810          2216  2 !                                                   sts_codes_desc_tbl) ;
; 1811          2217  2 !
; 1812          2218  2 ! Input Parameters:
; 1813          2219  2 !
; 1814          2220  2 !       Which_byte = diagnostic status code byte indicator.
; 1815          2221  2 !       Sts_codes_tbl = address of diagnostic status codes table.
; 1816          2222  2 !       Number_of_codes = number of sts codes in sts_codes_tbl.
; 1817          2223  2 !       Sts_codes_desc_tbl = address of table of descriptors containing text
; 1818          2224  2 !                            for the diagnostic sts codes.
; 1819          2225  2 !
; 1820          2226  2 ! Output Parameters:
; 1821          2227  2 !
; 1822          2228  2 !       None.
; 1823          2229  2 !
; 1824          2230  2 !--
; 1825          2231  2 Bind
; 1826          2232  2     Drive_sts = emb[82,0,8,0] : VECTOR [,byte],
; 1827          2233  2     Diag_sts_codes = .sts_codes_tbl : VECTOR [,byte],
; 1828          2234  2     Diag_sts_desc_tbl = .sts_codes_desc_tbl: BLOCKVECTOR [,long] ;
; 1829          2235  2
; 1830          2236  2 Arglist[0] = %ASCID 'UNKNOWN DIAGNOSTIC STS CODE' ;
; 1831          2237  2
; 1832          2238  2 Incr I from 0 to .number_of_codes do
; 1833          2239  3     Begin
; 1834          2240  3     If .drive_sts[.which_byte+7] EQL .diag_sts_codes[.I]
; 1835          2241  3     Then
; 1836          2242  3         Arglist[0] = diag_sts_desc_tbl[.I,0,0,0,0] ;
; 1837          2243  3         Exitloop ;
; 1838          2244  2     End ;
; 1839          2245  2
; 1840          2246  2 If .drive_sts[8] EQL 0 AND
; 1841          2247  2     .drive_sts[9] EQL 0
; 1842          2248  2 Then
; 1843          2249  2     Arglist[0] = %ASCID 'NO SOFT ERROR' ;
; 1844          2250  2
; 1845          2251  2 OUTPUT_LINES (  .fao_strings[3], arglist[0]  ) ;
; 1846          2252  2
; 1847          2253  2 Return ;
; 1848          2254  1 End ;   ! Routine


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

```
                                                        01E42                   .BLKB   2
53 4F 4E 47 41 49 44 20 4E 57 4F 4E 4B 4E 55            01E44 P.AQC:    .ASCII  \UNKNOWN DIAGNOSTIC STS CODE\<0>                 ;
      00 45 44 4F 43 20 53 54 53 20 43 49 54            01E53                                                                   ;
                                              010E001B  01E60 P.AQB:    .LONG   17694747                                      ;
                                              00000000' 01E64           .ADDRESS P.AQC                                        ;
00 00 52 4F 52 52 45 20 54 46 4F 53 20 4F 4E            01E68 P.AQE:    .ASCII  \NO SOFT ERROR\<0><0><0>                        ;
                                                 00     01E77                                                                 ;
                                              010E000D  01E78 P.AQD:    .LONG   17694733                                      ;
                                              00000000' 01E7C           .ADDRESS P.AQE                                        ;


                                                                          .PSECT  $CODE,NOWRT,  PIC,2

                                                    000C 00000 DIAGNOSTIC_STS_BYTE_DECODE:
                                                                          .WORD   Save R2,R3                                    ; 2203
                                    53 00000000G   00   9E 00002          MOVAB   DRIVE_STS, R3                                 ;
                                    52 00000000'   00   9E 00009          MOVAB   ARGLIST, R2                                   ;
                                    62 00000000'   00   9E 00010          MOVAB   P.AQB, ARGLIST                                ; 2236
                                    50             01   CE 00017          MNEGL   #1, I                                         ; 2238
                                                   1A   11 0001A          BRB     2$                                            ;
                                    51             63   9E 0001C 1$:      MOVAB   DRIVE_STS, R1                                 ; 2240
                                    51       04    AC   C0 0001F          ADDL2   WHICH_BYTE, R1                                ;
                                 08 BC40     07    A1   91 00023          CMPB    7(R1), @STS_CODES_TBL[I]                      ;
                                                   10   12 00029          BNEQ    3$                                            ;
                              51                   50   04   78 0002C     ASHL    #4, I, R1                                     ; 2242
                                    62       10 BC41   9E 0002F          MOVAB   @STS_CODES_DESC_TBL[R1], ARGLIST              ;
                                                   05   11 00034          BRB     3$                                            ; 2239
                              E1                   50   0C   AC   F3 00036 2$:     AOBLEQ  NUMBER_OF_CODES, I, 1$                ;
                                                   08   A3   95 00038 3$:     TSTB    DRIVE_STS+8                           ; 2246
                                                   0C   12 0003E          BNEQ    4$                                            ;
                                                   09   A3   95 00040          TSTB    DRIVE_STS+9                           ; 2247
                                                   07   12 00043          BNEQ    4$                                            ;
                                    62 00000000'   00   9E 00045          MOVAB   P.AQD, ARGLIST                                ; 2249
                                                   52   DD 0004C 4$:      PUSHL   R2                                            ; 2251
                                                   74   A2   DD 0004E          PUSHL   FAO_STRINGS+12                        ;
                                 00000000G  00          02   FB 00051          CALLS   #2, OUTPUT_LINES                      ;
                                                             04 00058          RET                                           ; 2254

; Routine Size:  89 bytes,    Routine Base:  $CODE + 0DD7

; 1849          2255  1
```

```
;  1851          2256  1 Routine CONVERT_BCD_NUMBER (bcd_number,number_of_digits) =
;  1852          2257  2 Begin
;  1853          2258  2
;  1854          2259  2 !++
;  1855          2260  2 !
;  1856          2261  2 ! Functional Description:
;  1857          2262  2 !
;  1858          2263  2 !    This routine will convert a BCD encoded number. It will return
;  1859          2264  2 !    the number in binary.
;  1860          2265  2 !
;  1861          2266  2 ! Calling Sequence:
;  1862          2267  2 !
;  1863          2268  2 !    Convert_bcd_number (bcd_number, number_of_digits) ;
;  1864          2269  2 !
;  1865          2270  2 ! Input Parameters:
;  1866          2271  2 !
;  1867          2272  2 !    Bcd_number = BCD number to be converted.
;  1868          2273  2 !    Number_of_digits = value, indicating the number of BCD digits to
;  1869          2274  2 !                       convert.
;  1870          2275  2 !
;  1871          2276  2 ! Output Parameters:
;  1872          2277  2 !
;  1873          2278  2 !    Binary value representing BCD encoded number.
;  1874          2279  2 !
;  1875          2280  2 !--
;  1876          2281  2
;  1877          2282  2 Local
;  1878          2283  2     Binary_value:       Initial (0),
;  1879          2284  2     Digit,
;  1880          2285  2     Offset:             Initial (0),
;  1881          2286  2     Result:             Initial (0) ;
;  1882          2287  2
;  1883          2288  2 Incr I from 0 to (.number_of_digits - 1) do
;  1884          2289  3     Begin
;  1885          2290  3     Digit = .bcd_number<.offset,4> ;
;  1886          2291  3     Result = OTS$POWJJ (10,.I) ;
;  1887          2292  3     Binary_value = .binary_value + (.digit * .result) ;
;  1888          2293  3     Offset = .offset + 4 ;
;  1889          2294  2     End ;
;  1890          2295  2
;  1891          2296  2
;  1892          2297  2 Return .binary_value ;
;  1893          2298  1 End ;   ! Routine



                                003C 00000 CONVERT_BCD_NUMBER:
                                           .WORD   Save R2,R3,R4,R5                    ; 2256
                                54  7C 00002        CLRQ    OFFSET                     ; 2257
                                50  D4 00004        CLRL    RESULT
                                                                                       ;
                          53    01  CE 00006        MNEGL   #1, I                      ; 2288
                                1B  11 00009        BRB     2$
                                                                                       ;
52        04    AC            54  EF 0000B 1$:     EXTZV   OFFSET, #4, BCD_NUMBER, DIGIT ; 2290
                                53  DD 00011        PUSHL   I                          ; 2291
                                0A  DD 00013        PUSHL   #10
                                                                                       ;
```

```
                   00000000G  00          02 FB 00015        CALLS   #2, OTS$POWJJ                           ;
              51              52          50 C5 0001C        MULL3   RESULT, DIGIT, R1                       ; 2292
                              55          51 C0 00020        ADDL2   R1, BINARY_VALUE                        ;
                              54          04 C0 00023        ADDL2   #4, OFFSET                              ; 2293
              E0              53    08    AC F2 00026 2$:    AOBLSS  NUMBER_OF_DIGITS, I, 1$                 ; 2288
                              50          55 D0 0002B        MOVL    BINARY_VALUE, R0                        ; 2297
                                             04 0002E        RET                                             ; 2298

; Routine Size:  47 bytes,    Routine Base:  $CODE + 0E30


; 1894          2299  1
; 1895          2300  1
; 1896          2301  1 End      ! Module
; 1897          2302  0 ELUDOM



;                               PSECT SUMMARY
;
;       Name                     Bytes                        Attributes
;
;  $OWN$                          1896  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  $PLIT                          7808  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  $CODE                          3679  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)



;                         Library Statistics
;
;                                          -------- Symbols --------      Pages      Processing
;       File                               Total    Loaded   Percent      Mapped     Time
;
;  _$255$DUA18:[SYSLIB]LIB.L32;1           18619      21         0        1000        00:01.5



;                                COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:TA78DVDEP/OBJ=OBJ$:TA78DVDEP MSRC$:TA78DVDEP/UPDATE=(BUG$:TA78DVDEP)

; Size:          3679 code + 9704 data bytes
; Run Time:         00:57.5
; Elapsed Time:     01:33.9
; Lines/CPU Min:     2401
; Lexemes/CPU-Min: 32415
; Memory Used:   552 pages
; Compilation Complete
```

0442 AH-EF71A-SE
VAX/VMS V4.1 SRC LST MCRF UPD

F11BXQP
MAP

ACCESS
LIS

TA78DVDEP
LIS

CLENUP
LIS

CHARGEQ
LIS

F11X