```
DDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBB      UUU         UUU     GGGGGGGGGGG
DDDDDDDDDDDD    EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUU         UUU     GGGGGGGGGGGG
DDDDDDDDDDDD    EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUU         UUU     GGGGGGGGGGGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG
DDD         DDD EEEEEEEEEEE      BBBBBBBBBBBB     UUU         UUU GGG
DDD         DDD EEEEEEEEEEE      BBBBBBBBBBBB     UUU         UUU GGG
DDD         DDD EEEEEEEEEEE      BBBBBBBBBBBB     UUU         UUU GGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG   GGGGGGGGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG   GGGGGGGGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG   GGGGGGGGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG         GGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG         GGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG         GGG
DDDDDDDDDDDD    EEEEEEEEEEEEEEE  BBBBBBBBBBBB      UUUUUUUUUUUUUUUUU     GGGGGGGGG
DDDDDDDDDDDD    EEEEEEEEEEEEEEE  BBBBBBBBBBBB      UUUUUUUUUUUUUUUUU     GGGGGGGGG
DDDDDDDDDDDD    EEEEEEEEEEEEEEE  BBBBBBBBBBBB      UUUUUUUUUUUUUUUUU     GGGGGGGGG
```

FILEID**DBGENCDEC

```
DDDDDDDD    BBBBBBBB      GGGGGGGG  EEEEEEEEEE  NN      NN    CCCCCCCC  DDDDDDDD    EEEEEEEEEE    CCCCCCCC
DDDDDDDD    BBBBBBBB      GGGGGGGG  EEEEEEEEEE  NN      NN    CCCCCCCC  DDDDDDDD    EEEEEEEEEE    CCCCCCCC
DD      DD  BB      BB  GG          EE          NN      NN  CC          DD      DD  EE          CC
DD      DD  BB      BB  GG          EE          NN      NN  CC          DD      DD  EE          CC
DD      DD  BB      BB  GG          EE          NNNN    NN  CC          DD      DD  EE          CC
DD      DD  BB      BB  GG          EE          NNNN    NN  CC          DD      DD  EE          CC
DD      DD  BBBBBBBB    GG          EEEEEEEE    NN  NN  NN  CC          DD      DD  EEEEEEEE    CC
DD      DD  BBBBBBBB    GG          EEEEEEEE    NN  NN  NN  CC          DD      DD  EEEEEEEE    CC
DD      DD  BB      BB  GG  GGGGGG  EE          NN    NNNN  CC          DD      DD  EE          CC
DD      DD  BB      BB  GG  GGGGGG  EE          NN    NNNN  CC          DD      DD  EE          CC
DD      DD  BB      BB  GG      GG  EE          NN      NN  CC          DD      DD  EE          CC                ....
DD      DD  BB      BB  GG      GG  EE          NN      NN  CC          DD      DD  EE          CC                ....
DDDDDDDD    BBBBBBBB      GGGGGG    EEEEEEEEEE  NN      NN    CCCCCCCC  DDDDDDDD    EEEEEEEEEE    CCCCCCCC      ....
DDDDDDDD    BBBBBBBB      GGGGGG    EEEEEEEEEE  NN      NN    CCCCCCCC  DDDDDDDD    EEEEEEEEEE    CCCCCCCC      ....
```

```
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
```

```
;     1          0001  0 MODULE DBGENCDEC (IDENT = 'V04-000') =
;     2          0002  1 BEGIN
;     3          0003  1
;     4          0004  1 !**************************************************************************
;     5          0005  1 !*                                                                        *
;     6          0006  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                              *
;     7          0007  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                *
;     8          0008  1 !*  ALL RIGHTS RESERVED.                                                  *
;     9          0009  1 !*                                                                        *
;    10          0010  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
;    11          0011  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE *
;    12          0012  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER *
;    13          0013  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
;    14          0014  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY *
;    15          0015  1 !*  TRANSFERRED.                                                           *
;    16          0016  1 !*                                                                        *
;    17          0017  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE *
;    18          0018  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT *
;    19          0019  1 !*  CORPORATION.                                                           *
;    20          0020  1 !*                                                                        *
;    21          0021  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS *
;    22          0022  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
;    23          0023  1 !*                                                                        *
;    24          0024  1 !*                                                                        *
;    25          0025  1 !**************************************************************************
;    26          0026  1
;    27          0027  1 !++
;    28          0028  1 !
;    29          0029  1 !       Original Author:  John Francis
;    30          0030  1 !
;    31          0031  1 !       Modification history:
;    32          0032  1 !
;    33          0033  1 !       001     Walter Carrell III, 3-Jun-83
;    34          0034  1 !               ASHP thought it had only 5 arguments. See the comment for
;    35          0035  1 !               the MACRO Opcode_list for a more complete explination.
;    36          0036  1 !
;    37          0037  1 !       002     Walter Carrell III, 08-Jun-83
;    38          0038  1 !               The outside world expects DBG$OPCODE_INDEX to return an index
;    39          0039  1 !               into DBG$Opcode_Kind_Table.  Within DBGENCDEC DBG$OPCODE_INDEX
;    40          0040  1 !               was expected to return an index into DBG$Opcode_Name_Table.
;    41          0041  1 !               DBG$OPCODE_INDEX originally returned an index into
;    42          0042  1 !               DBG$Opcode_Name_Table.  This edit changes that.  The place
;    43          0043  1 !               where DBG$OPCODE_INDEX was called within DBGENCDEC have been
;    44          0044  1 !               changed to call Opcode_Name_Index, a local routine which
;    45          0045  1 !               is the original DBG$OPCODE_INDEX with some fixed to make
;    46          0046  1 !               it work correctly.  A new DBG$OPCODE_INDEX was written to
;    47          0047  1 !               return an index into DBG$Opcode_Kind_table.
;    48          0048  1 !
;    49          0049  1 !       003     Walter Carrell III, 09-Jun-83
;    50          0050  1 !               1. CVTTP was out of order in the Mnemonic table.
;    51          0051  1 !               2. Allow ^Y out of the printing of destinations of a CASE list.
;    52          0052  1 !               3. The limit of CASE statements was always being read as a LONG
;    53          0053  1 !                  instead of in the appropriate type.
;    54          0054  1 !               4. XFC and BUGx had a bad table entrys.
;    55          0055  1 !
;    56          0056  1 !       004     Walter Carrell III, 13-Jun-83
;    57          0057  1 !               All the 2 byte opcode instructions had FE instead of FD in the
```

```
;    58          0058  1 !                   table for the first byte of the opcode.
;    59          0059  1 !
;    60          0060  1 !           005     Walter Carrell III, 20-Jun-83
;    61          0061  1 !                   1. ^S and ^I were not allowed
;    62          0062  1 !                   2. Addr-exprs were not allowed in any context other that
;    63          0063  1 !                      branch displacements
;    64          0064  1 !                   3. (Rn)[Rn] where Rn is the same for both was not caught
;    65          0065  1 !                   4. Choose displacement sizes instead of defaulting to Byte.
;    66          0066  1 !                   5. Allow SP in indexed mode, [SP]
;    67          0067  1 !                   6. Correct the CASE offset address calculation
;    68          0068  1 !                   7. Correct the context of ASHP
;    69          0069  1 !                   8. Make DEP/INSTR for I^# work
;    70          0070  1 !
;    71          0071  1 !           006     Walter Carrell III, 11-Jul-83
;    72          0072  1 !                   Fix Parse_Register so it doesn't use Parse_Expression
;    73          0073  1 !                   Parse_Expression cannot be used because a scope may
;    74          0074  1 !                   not be active the a DEP/INSTR is issued and therefore
;    75          0075  1 !                   Rn is thought to be a symbol outside the active scope.
;    76          0076  1 !
;    77          0077  1 !           007     Walter Carrell III, 08-Aug-83
;    78          0078  1 !                   Enhance error reporting of instruction Encoding
;    79          0079  1 !
;    80          0080  1 !           008     Walter Carrell III, 20-Sep-83
;    81          0081  1 !                   Fix instruction encoding to allow quad and octaword literals
;    82          0082  1 !
;    83          0083  1 !--                                                                    !
;    84          0084  1
;    85          0085  1 REQUIRE 'SRC$:DBGPROLOG.REQ';
;    86          0219  1
;    87          0220  1 FORWARD ROUTINE
;    88          0221  1 !
;    89          0222  1 ! Global Routines
;    90          0223  1 !
;    91          0224  1     DBG$Ins_Decode,                 ! Decode Single Instruction
;    92          0225  1     DBG$Ins_Encode,                 ! Encode Single instruction
;    93          0226  1     DBG$Opcode_Index,               ! Convert Mnemonic to kind table index     ! Changed to return Kind tabl
;    94          0227  1                                                                            !instead of name table index.
;    95          0228  1 !
;    96          0229  1 ! Local Routines
;    97          0230  1 !
;    98          0231  1     Opcode_Name_Index,              ! Convert Mnemonic to name table index     !
;    99          0232  1     Fetch_Instruction   : NOVALUE,  ! Fetch bytes from instrction stream
;   100          0233  1     Fetch_Operand       : NOVALUE,  ! Fetch (and optionally print) an operand
;   101          0234  1     Print_Address       : NOVALUE,  ! Print operand address
;   102          0235  1     Print_Operand       : NOVALUE,  ! Print operand value
;   103          0236  1     Parse_Operand       : NOVALUE,  ! Parse one instruction operand
;   104          0237  1     Parse_Expression,               ! Parse an operand (address or value)
;   105          0238  1     Parse_Register,                 ! Parse a register name
;   106          0239  1     Check_Register,                 ! See if address describes a register
;   107          0240  1     Store_Operand       : NOVALUE,  ! Store bytes in output stream
;   108          0241  1     Scan_Operand,                   ! Separate one operand string
;   109          0242  1     Skip_Leading_Blanks : NOVALUE;  ! Skip over leading spaces and/or tabs
;   110          0243  1
;   111          0244  1 EXTERNAL
;   112          0245  1     DBG$GB_RADIX: VECTOR[3, BYTE];      ! Radix settings
;   113          0246  1
;   114          0247  1
```

```
;  115          0248  1 EXTERNAL ROUTINE
;  116          0249  1     DBG$CONV_TEXT_VALUE,                                          !
;  117          0250  1     DBG$COVER_DX_DX,                                              !
;  118          0251  1     DBG$Print                       : NOVALUE,
;  119          0252  1     DBG$Print_Value                 : NOVALUE,
;  120          0253  1     DBG$Print_Identifier_PC         : NOVALUE,
;  121          0254  1     DBG$NewLine                     : NOVALUE,
;  122          0255  1     DBG$Pop_Tempmem                 : NOVALUE,
;  123          0256  1     DBG$Push_Tempmem,
;  124          0257  1     DBG$Is_It_Entry,
;  125          0258  1     DBG$Make_Val_Desc,
;  126          0259  1     DBG$Nparse_Address,
;  127          0260  1     DBG$Nparse_Expression,
;  128          0261  1     DBG$Prim_to_Val;
;  129          0262  1
;  130          0263  1 LITERAL
;  131          0264  1     simple_0_operand    = %X'00',
;  132          0265  1     simple_1_operand    = %X'01',
;  133          0266  1     simple_2_operand    = %X'02',
;  134          0267  1     simple_3_operand    = %X'03',
;  135          0268  1     branch_0_operand    = %X'04',
;  136          0269  1     branch_1_operand    = %X'05',
;  137          0270  1     branch_2_operand    = %X'06',
;  138          0271  1     branch_3_operand    = %X'07',
;  139          0272  1     convert_datatype    = %X'08',
;  140          0273  1     evaluate_address    = %X'08',
;  141          0274  1     simple_bit_field    = %X'08',
;  142          0275  1     routine_dispatch    = %X'08',
;  143          0276  1     locate_character    = %X'08',
;  144          0277  1     polynomial_value    = %X'08',
;  145          0278  1     probe_for_access    = %X'08',
;  146          0279  1     trailing_operand    = %X'09',
;  147          0280  1     string_3_operand    = %X'0A',
;  148          0281  1     string_4_operand    = %X'0B',
;  149          0282  1     string_5_operand    = %X'0C',
;  150          0283  1     string_6_operand    = %X'0D',
;  151          0284  1
;  152          0285  1     complex_SHIFT       = %X'10',
;  153          0286  1     complex_CASE        = %X'11',
;  154          0287  1     complex_EDIV        = %X'12',
;  155          0288  1     complex_EMOD        = %X'13',
;  156          0289  1     complex_EMUL        = %X'14',
;  157          0290  1     complex_INDEX       = %X'15',
;  158          0291  1     complex_CRC         = %X'16',
;  159          0292  1     complex_ASHP        = %X'17',
;  160          0293  1
;  161          0294  1     maximum_state       = %X'17',
;  162          0295  1
;  163          0296  1     context_b           = %X'00',
;  164          0297  1     context_w           = %X'01',
;  165          0298  1     context_l           = %X'02',
;  166          0299  1     context_q           = %X'03',
;  167          0300  1     context_o           = %X'04',
;  168          0301  1     context_f           = %X'05',
;  169          0302  1     context_d           = %X'06',
;  170          0303  1     context_g           = %X'07',
;  171          0304  1     context_h           = %X'08',
```

```
;  172     0305  1     context_bu      = %X'09',
;  173     0306  1     context_wu      = %X'0A',
;  174     0307  1     context_t       = %X'0B',       ! size.wu       base.b
;  175     0308  1     context_p       = %X'0C',       ! size.wu       base.b
;  176     0309  1     context_m       = %X'0D',       !  pos.l        size.b          base.b
;  177     0310  1     context_v       = %X'0E';       !  pos.l        base.b
```

```
;  179     0311  1 !++
;  180     0312  1 !       The following table is used to build 2 data structures:
;  181     0313  1 !
;  182     0314  1 !       DBG$Opcode_Name_Table - An alphabetical table of the Opcode names
;  183     0315  1 !       DBG$Opcode_Kind_table - A back translation table to get from an
;  184     0316  1 !                               Op code to the Name.
;  185     0317  1 !
;  186     0318  1 !       Opcode_entry is a macro the is defined twice to pass over
;  187     0319  1 !       Opcode_List twice to buld the two tables.
;  188     0320  1 !
;  189     0321  1 !       The arguments have the following definition:
;  190     0322  1 !
;  191     0323  1 !       1. The first argument is a flag that indicates that the entry
;  192     0324  1 !          has a duplicate Opcode and that it should be ignored in
;  193     0325  1 !          DBG$Opcode_Kind_Table.
;  194     0326  1 !       2. The second argument is the MNEMONIC.  It must be 6 characters.
;  195     0327  1 !       3. The third argument is the Opcode.
;  196     0328  1 !       4. The fourth argument is the state to start with in the finite
;  197     0329  1 !          state machine
;  198     0330  1 !       5. The fifth and optional sixth arguments are context flags.
;  199     0331  1 !          Their nature is not fully understood.
;  200     0332  1 !
;  201     0333  1 !       Note that the table must be in alphabetical order by MNEMONIC.
;  202     0334  1 !       A binary search is used to find table entries.
;  203     0335  1 !
;  204     0336  1 !       The editorial starts here.
;  205     0337  1 !
;  206     0338  1 !       The table is more complex than necessary for the simple matter
;  207     0339  1 !       of decoding for output and encoding instructions for deposit.
;  208     0340  1 !       The number of arguments would have been sufficient, instead of
;  209     0341  1 !       the last 3 arguments. The first byte of an argument tells you
;  210     0342  1 !       what you need to know about the rest of the argument.
;  211     0343  1 !
;  212     0344  1 !       The intent was apparently to have enough information in the
;  213     0345  1 !       table to allow the decoding of the instructions for interperting
;  214     0346  1 !       watch points in the stack and registers.  The information in the
;  215     0347  1 !       table is not sufficient for that purpose.
;  216     0348  1 !
;  217     0349  1 !--
;  218   M 0350  1 MACRO Opcode_List =
;  219   M 0351  1         Opcode_Entry(1,'ACBB  ',%X'9D'  ,branch_3_operand,context_b,context_w),
;  220   M 0352  1         Opcode_Entry(1,'ACBD  ',%X'6F'  ,branch_3_operand,context_d,context_w),
;  221   M 0353  1         Opcode_Entry(1,'ACBF  ',%X'4F'  ,branch_3_operand,context_f,context_w),
;  222   M 0354  1         Opcode_Entry(1,'ACBG  ',%X'4FFD',branch_3_operand,context_g,context_w),
;  223   M 0355  1         Opcode_Entry(1,'ACBH  ',%X'6FFD',branch_3_operand,context_h,context_w),
;  224   M 0356  1         Opcode_Entry(1,'ACBL  ',%X'F1'  ,branch_3_operand,context_l,context_w),
;  225   M 0357  1         Opcode_Entry(1,'ACBW  ',%X'3D'  ,branch_3_operand,context_w,context_w),
;  226   M 0358  1         Opcode_Entry(1,'ADAWI ',%X'58'  ,simple_2_operand,context_w),
;  227   M 0359  1         Opcode_Entry(1,'ADDB2 ',%X'80'  ,simple_2_operand,context_b),
;  228   M 0360  1         Opcode_Entry(1,'ADDB3 ',%X'81'  ,simple_3_operand,context_b),
;  229   M 0361  1         Opcode_Entry(1,'ADDD2 ',%X'60'  ,simple_2_operand,context_d),
;  230   M 0362  1         Opcode_Entry(1,'ADDD3 ',%X'61'  ,simple_3_operand,context_d),
;  231   M 0363  1         Opcode_Entry(1,'ADDF2 ',%X'40'  ,simple_2_operand,context_f),
;  232   M 0364  1         Opcode_Entry(1,'ADDF3 ',%X'41'  ,simple_3_operand,context_f),
;  233   M 0365  1         Opcode_Entry(1,'ADDG2 ',%X'40FD',simple_2_operand,context_g),
;  234   M 0366  1         Opcode_Entry(1,'ADDG3 ',%X'41FD',simple_3_operand,context_g),
;  235   M 0367  1         Opcode_Entry(1,'ADDH2 ',%X'60FD',simple_2_operand,context_h),
```

```
;  236        M 0368  1          Opcode_Entry(1,'ADDH3 ',%X'61FD',simple_3_operand,context_h),          !
;  237        M 0369  1          Opcode_Entry(1,'ADDL2 ',%X'C0'  ,simple_2_operand,context_l),
;  238        M 0370  1          Opcode_Entry(1,'ADDL3 ',%X'C1'  ,simple_3_operand,context_l),
;  239        M 0371  1          Opcode_Entry(1,'ADDP4 ',%X'20'  ,simple_2_operand,context_p),
;  240        M 0372  1          Opcode_Entry(1,'ADDP6 ',%X'21'  ,simple_3_operand,context_p),
;  241        M 0373  1          Opcode_Entry(1,'ADDW2 ',%X'A0'  ,simple_2_operand,context_w),
;  242        M 0374  1          Opcode_Entry(1,'ADDW3 ',%X'A1'  ,simple_3_operand,context_w),
;  243        M 0375  1          Opcode_Entry(1,'ADWC  ',%X'D8'  ,simple_2_operand,context_l),
;  244        M 0376  1          Opcode_Entry(1,'AOBLEQ',%X'F3'  ,branch_2_operand,context_l,context_b),
;  245        M 0377  1          Opcode_Entry(1,'AOBLSS',%X'F2'  ,branch_2_operand,context_l,context_b),
;  246        M 0378  1          Opcode_Entry(1,'ASHL  ',%X'78'  ,complex_SHIFT   ,context_l),
;  247        M 0379  1          Opcode_Entry(1,'ASHP  ',%X'F8'  ,complex_ASHP,context_b,context_p),     ! simple_3_operand to comple
;  248        M 0380  1          Opcode_Entry(1,'ASHQ  ',%X'79'  ,complex_SHIFT   ,context_q),
;  249        M 0381  1          Opcode_Entry(1,'BBC   ',%X'E1'  ,branch_1_operand,context_v,context_b),
;  250        M 0382  1          Opcode_Entry(1,'BBCC  ',%X'E5'  ,branch_1_operand,context_v,context_b),
;  251        M 0383  1          Opcode_Entry(1,'BBCCI ',%X'E7'  ,branch_1_operand,context_v,context_b),
;  252        M 0384  1          Opcode_Entry(1,'BBCS  ',%X'E3'  ,branch_1_operand,context_v,context_b),
;  253        M 0385  1          Opcode_Entry(1,'BBS   ',%X'E0'  ,branch_1_operand,context_v,context_b),
;  254        M 0386  1          Opcode_Entry(1,'BBSC  ',%X'E4'  ,branch_1_operand,context_v,context_b),
;  255        M 0387  1          Opcode_Entry(1,'BBSS  ',%X'E2'  ,branch_1_operand,context_v,context_b),
;  256        M 0388  1          Opcode_Entry(1,'BBSSI ',%X'E6'  ,branch_1_operand,context_v,context_b),
;  257        M 0389  1          Opcode_Entry(0,'BCC   ',%X'1E'  ,branch_0_operand,context_b),
;  258        M 0390  1          Opcode_Entry(0,'BCS   ',%X'1F'  ,branch_0_operand,context_b),
;  259        M 0391  1          Opcode_Entry(1,'BEQL  ',%X'13'  ,branch_0_operand,context_b),
;  260        M 0392  1          Opcode_Entry(0,'BEQLU ',%X'13'  ,branch_0_operand,context_b),
;  261        M 0393  1          Opcode_Entry(1,'BGEQ  ',%X'18'  ,branch_0_operand,context_b),
;  262        M 0394  1          Opcode_Entry(1,'BGEQU ',%X'1E'  ,branch_0_operand,context_b),
;  263        M 0395  1          Opcode_Entry(1,'BGTR  ',%X'14'  ,branch_0_operand,context_b),
;  264        M 0396  1          Opcode_Entry(1,'BGTRU ',%X'1A'  ,branch_0_operand,context_b),
```

```
;  266     M 0397  1          Opcode_Entry(1,'BICB2  ',%X'8A'  ,simple_2_operand,context_b),
;  267     M 0398  1          Opcode_Entry(1,'BICB3  ',%X'8B'  ,simple_3_operand,context_b),
;  268     M 0399  1          Opcode_Entry(1,'BICL2  ',%X'CA'  ,simple_2_operand,context_l),
;  269     M 0400  1          Opcode_Entry(1,'BICL3  ',%X'CB'  ,simple_3_operand,context_l),
;  270     M 0401  1          Opcode_Entry(1,'BICPSW ',%X'B9'  ,simple_1_operand,context_w),
;  271     M 0402  1          Opcode_Entry(1,'BICW2  ',%X'AA'  ,simple_2_operand,context_w),
;  272     M 0403  1          Opcode_Entry(1,'BICW3  ',%X'AB'  ,simple_3_operand,context_w),
;  273     M 0404  1          Opcode_Entry(1,'BISB2  ',%X'88'  ,simple_2_operand,context_b),
;  274     M 0405  1          Opcode_Entry(1,'BISB3  ',%X'89'  ,simple_3_operand,context_b),
;  275     M 0406  1          Opcode_Entry(1,'BISL2  ',%X'C8'  ,simple_2_operand,context_l),
;  276     M 0407  1          Opcode_Entry(1,'BISL3  ',%X'C9'  ,simple_3_operand,context_l),
;  277     M 0408  1          Opcode_Entry(1,'BISPSW ',%X'B8'  ,simple_1_operand,context_w),
;  278     M 0409  1          Opcode_Entry(1,'BISW2  ',%X'A8'  ,simple_2_operand,context_w),
;  279     M 0410  1          Opcode_Entry(1,'BISW3  ',%X'A9'  ,simple_3_operand,context_w),
;  280     M 0411  1          Opcode_Entry(1,'BITB   ',%X'93'  ,simple_2_operand,context_b),
;  281     M 0412  1          Opcode_Entry(1,'BITL   ',%X'D3'  ,simple_2_operand,context_l),
;  282     M 0413  1          Opcode_Entry(1,'BITW   ',%X'B3'  ,simple_2_operand,context_w),
;  283     M 0414  1          Opcode_Entry(1,'BLBC   ',%X'E9'  ,branch_1_operand,context_l,context_b),
;  284     M 0415  1          Opcode_Entry(1,'BLBS   ',%X'E8'  ,branch_1_operand,context_l,context_b),
;  285     M 0416  1          Opcode_Entry(1,'BLEQ   ',%X'15'  ,branch_0_operand,context_b),
;  286     M 0417  1          Opcode_Entry(1,'BLEQU  ',%X'1B'  ,branch_0_operand,context_b),
;  287     M 0418  1          Opcode_Entry(1,'BLSS   ',%X'19'  ,branch_0_operand,context_b),
;  288     M 0419  1          Opcode_Entry(1,'BLSSU  ',%X'1F'  ,branch_0_operand,context_b),
;  289     M 0420  1          Opcode_Entry(1,'BNEQ   ',%X'12'  ,branch_0_operand,context_b),
;  290     M 0421  1          Opcode_Entry(0,'BNEQU  ',%X'12'  ,branch_0_operand,context_b),
;  291     M 0422  1          Opcode_Entry(1,'BPT    ',%X'03'  ,simple_0_operand),
;  292     M 0423  1          Opcode_Entry(1,'BRB    ',%X'11'  ,branch_0_operand,context_b),
;  293     M 0424  1          Opcode_Entry(1,'BRW    ',%X'31'  ,branch_0_operand,context_w),
;  294     M 0425  1          Opcode_Entry(1,'BSBB   ',%X'10'  ,branch_0_operand,context_b),
;  295     M 0426  1          Opcode_Entry(1,'BSBW   ',%X'30'  ,branch_0_operand,context_w),
;  296     M 0427  1          Opcode_Entry(1,'BUGL   ',%X'FDFF',simple_0_operand),
;  297     M 0428  1          Opcode_Entry(1,'BUGW   ',%X'FEFF',simple_0_operand),
;  298     M 0429  1          Opcode_Entry(1,'BVC    ',%X'1C'  ,branch_0_operand,context_b),
;  299     M 0430  1          Opcode_Entry(1,'BVS    ',%X'1D'  ,branch_0_operand,context_b),
;  300     M 0431  1          Opcode_Entry(1,'CALLG  ',%X'FA'  ,routine_dispatch,context_b,context_b),
;  301     M 0432  1          Opcode_Entry(1,'CALLS  ',%X'FB'  ,routine_dispatch,context_l,context_b),
;  302     M 0433  1          Opcode_Entry(1,'CASEB  ',%X'8F'  ,complex_CASE    ,context_b),
;  303     M 0434  1          Opcode_Entry(1,'CASEL  ',%X'CF'  ,complex_CASE    ,context_l),
;  304     M 0435  1          Opcode_Entry(1,'CASEW  ',%X'AF'  ,complex_CASE    ,context_w),
;  305     M 0436  1          Opcode_Entry(1,'CHME   ',%X'BD'  ,simple_1_operand,context_w),
;  306     M 0437  1          Opcode_Entry(1,'CHMK   ',%X'BC'  ,simple_1_operand,context_w),
;  307     M 0438  1          Opcode_Entry(1,'CHMS   ',%X'BE'  ,simple_1_operand,context_w),
;  308     M 0439  1          Opcode_Entry(1,'CHMU   ',%X'BF'  ,simple_1_operand,context_w),
;  309     M 0440  1          Opcode_Entry(1,'CLRB   ',%X'94'  ,simple_1_operand,context_b),
;  310     M 0441  1          Opcode_Entry(0,'CLRD   ',%X'7C'  ,simple_1_operand,context_d),
;  311     M 0442  1          Opcode_Entry(0,'CLRF   ',%X'D4'  ,simple_1_operand,context_f),
;  312     M 0443  1          Opcode_Entry(0,'CLRG   ',%X'7C'  ,simple_1_operand,context_g),
;  313     M 0444  1          Opcode_Entry(0,'CLRH   ',%X'7CFD',simple_1_operand,context_h),          !
;  314     M 0445  1          Opcode_Entry(1,'CLRL   ',%X'D4'  ,simple_1_operand,context_l),
;  315     M 0446  1          Opcode_Entry(1,'CLRO   ',%X'7CFD',simple_1_operand,context_o),          !
;  316     M 0447  1          Opcode_Entry(1,'CLRQ   ',%X'7C'  ,simple_1_operand,context_q),
;  317     M 0448  1          Opcode_Entry(1,'CLRW   ',%X'B4'  ,simple_1_operand,context_w),
```

```
;  319     M 0449  1       Opcode_Entry(1,'CMPB  ',%X'91'   ,simple_2_operand,context_b),
;  320     M 0450  1       Opcode_Entry(1,'CMPC3 ',%X'29'   ,string_3_operand,context_t,context_b),
;  321     M 0451  1       Opcode_Entry(1,'CMPC5 ',%X'2D'   ,string_5_operand,context_t,context_t),
;  322     M 0452  1       Opcode_Entry(1,'CMPD  ',%X'71'   ,simple_2_operand,context_d),
;  323     M 0453  1       Opcode_Entry(1,'CMPF  ',%X'51'   ,simple_2_operand,context_f),
;  324     M 0454  1       Opcode_Entry(1,'CMPG  ',%X'51FD',simple_2_operand,context_g),          !
;  325     M 0455  1       Opcode_Entry(1,'CMPH  ',%X'71FD',simple_2_operand,context_h),          !
;  326     M 0456  1       Opcode_Entry(1,'CMPL  ',%X'D1'   ,simple_2_operand,context_l),
;  327     M 0457  1       Opcode_Entry(1,'CMPP3 ',%X'35'   ,string_3_operand,context_p,context_b),
;  328     M 0458  1       Opcode_Entry(1,'CMPP4 ',%X'37'   ,simple_2_operand,context_p),
;  329     M 0459  1       Opcode_Entry(1,'CMPV  ',%X'EC'   ,simple_bit_field,context_m,context_l),
;  330     M 0460  1       Opcode_Entry(1,'CMPW  ',%X'B1'   ,simple_2_operand,context_w),
;  331     M 0461  1       Opcode_Entry(1,'CMPZV ',%X'ED'   ,simple_bit_field,context_m,context_l),
;  332     M 0462  1       Opcode_Entry(1,'CRC   ',%X'0B'   ,complex_CRC),
;  333     M 0463  1       Opcode_Entry(1,'CVTBD ',%X'6C'   ,convert_datatype,context_b,context_d),
;  334     M 0464  1       Opcode_Entry(1,'CVTBF ',%X'4C'   ,convert_datatype,context_b,context_f),
;  335     M 0465  1       Opcode_Entry(1,'CVTBG ',%X'4CFD',convert_datatype,context_b,context_g),!
;  336     M 0466  1       Opcode_Entry(1,'CVTBH ',%X'6CFD',convert_datatype,context_b,context_h),!
;  337     M 0467  1       Opcode_Entry(1,'CVTBL ',%X'98'   ,convert_datatype,context_b,context_l),
;  338     M 0468  1       Opcode_Entry(1,'CVTBW ',%X'99'   ,convert_datatype,context_b,context_w),
;  339     M 0469  1       Opcode_Entry(1,'CVTDB ',%X'68'   ,convert_datatype,context_d,context_b),
;  340     M 0470  1       Opcode_Entry(1,'CVTDF ',%X'76'   ,convert_datatype,context_d,context_f),
;  341     M 0471  1       Opcode_Entry(1,'CVTDH ',%X'32FD',convert_datatype,context_d,context_h),!
;  342     M 0472  1       Opcode_Entry(1,'CVTDL ',%X'6A'   ,convert_datatype,context_d,context_l),
;  343     M 0473  1       Opcode_Entry(1,'CVTDW ',%X'69'   ,convert_datatype,context_d,context_w),
;  344     M 0474  1       Opcode_Entry(1,'CVTFB ',%X'48'   ,convert_datatype,context_f,context_b),
;  345     M 0475  1       Opcode_Entry(1,'CVTFD ',%X'56'   ,convert_datatype,context_f,context_d),
;  346     M 0476  1       Opcode_Entry(1,'CVTFG ',%X'99FD',convert_datatype,context_f,context_g),!
;  347     M 0477  1       Opcode_Entry(1,'CVTFH ',%X'98FD',convert_datatype,context_f,context_h),!
;  348     M 0478  1       Opcode_Entry(1,'CVTFL ',%X'4A'   ,convert_datatype,context_f,context_l),
;  349     M 0479  1       Opcode_Entry(1,'CVTFW ',%X'49'   ,convert_datatype,context_f,context_w),
;  350     M 0480  1       Opcode_Entry(1,'CVTGB ',%X'48FD',convert_datatype,context_g,context_b),!
;  351     M 0481  1       Opcode_Entry(1,'CVTGF ',%X'33FD',convert_datatype,context_g,context_f),!
;  352     M 0482  1       Opcode_Entry(1,'CVTGH ',%X'56FD',convert_datatype,context_g,context_h),!
;  353     M 0483  1       Opcode_Entry(1,'CVTGL ',%X'4AFD',convert_datatype,context_g,context_l),!
;  354     M 0484  1       Opcode_Entry(1,'CVTGW ',%X'49FD',convert_datatype,context_g,context_w),!
;  355     M 0485  1       Opcode_Entry(1,'CVTHB ',%X'68FD',convert_datatype,context_h,context_b),!
;  356     M 0486  1       Opcode_Entry(1,'CVTHD ',%X'F7FD',convert_datatype,context_h,context_d),!
;  357     M 0487  1       Opcode_Entry(1,'CVTHF ',%X'F6FD',convert_datatype,context_h,context_f),!
;  358     M 0488  1       Opcode_Entry(1,'CVTHG ',%X'76FD',convert_datatype,context_h,context_g),!
;  359     M 0489  1       Opcode_Entry(1,'CVTHL ',%X'6AFD',convert_datatype,context_h,context_l),!
;  360     M 0490  1       Opcode_Entry(1,'CVTHW ',%X'69FD',convert_datatype,context_h,context_w),!
;  361     M 0491  1       Opcode_Entry(1,'CVTLB ',%X'F6'   ,convert_datatype,context_l,context_b),
;  362     M 0492  1       Opcode_Entry(1,'CVTLD ',%X'6E'   ,convert_datatype,context_l,context_d),
;  363     M 0493  1       Opcode_Entry(1,'CVTLF ',%X'4E'   ,convert_datatype,context_l,context_f),
;  364     M 0494  1       Opcode_Entry(1,'CVTLG ',%X'4EFD',convert_datatype,context_l,context_g),!
;  365     M 0495  1       Opcode_Entry(1,'CVTLH ',%X'6EFD',convert_datatype,context_l,context_h),!
;  366     M 0496  1       Opcode_Entry(1,'CVTLP ',%X'F9'   ,convert_datatype,context_l,context_p),
;  367     M 0497  1       Opcode_Entry(1,'CVTLW ',%X'F7'   ,convert_datatype,context_l,context_w),
```

```
;  369        M 0498  1        Opcode_Entry(1,'CVTPL ',%X'36'   ,convert_datatype,context_p,context_l),
;  370        M 0499  1        Opcode_Entry(1,'CVTPS ',%X'08'   ,convert_datatype,context_p,context_t),
;  371        M 0500  1        Opcode_Entry(1,'CVTPT ',%X'24'   ,string_5_operand,context_p,context_t),
;  372        M 0501  1        Opcode_Entry(1,'CVTRDL',%X'6B'   ,convert_datatype,context_d,context_l),
;  373        M 0502  1        Opcode_Entry(1,'CVTRFL',%X'4B'   ,convert_datatype,context_f,context_l),
;  374        M 0503  1        Opcode_Entry(1,'CVTRGL',%X'4BFD',convert_datatype,context_g,context_l),!
;  375        M 0504  1        Opcode_Entry(1,'CVTRHL',%X'6BFD',convert_datatype,context_h,context_l),!
;  376        M 0505  1        Opcode_Entry(1,'CVTSP ',%X'09'   ,convert_datatype,context_t,context_p),
;  377        M 0506  1        Opcode_Entry(1,'CVTTP ',%X'26'   ,string_5_operand,context_t,context_p),!
;  378        M 0507  1        Opcode_Entry(1,'CVTWB ',%X'33'   ,convert_datatype,context_w,context_b),
;  379        M 0508  1        Opcode_Entry(1,'CVTWD ',%X'6D'   ,convert_datatype,context_w,context_d),
;  380        M 0509  1        Opcode_Entry(1,'CVTWF ',%X'4D'   ,convert_datatype,context_w,context_f),
;  381        M 0510  1        Opcode_Entry(1,'CVTWG ',%X'4DFD',convert_datatype,context_w,context_g),!
;  382        M 0511  1        Opcode_Entry(1,'CVTWH ',%X'6DFD',convert_datatype,context_w,context_h),!
;  383        M 0512  1        Opcode_Entry(1,'CVTWL ',%X'32'   ,convert_datatype,context_w,context_l),
;  384        M 0513  1        Opcode_Entry(1,'DECB  ',%X'97'   ,simple_1_operand,context_b),
;  385        M 0514  1        Opcode_Entry(1,'DECL  ',%X'D7'   ,simple_1_operand,context_l),
;  386        M 0515  1        Opcode_Entry(1,'DECW  ',%X'B7'   ,simple_1_operand,context_w),
;  387        M 0516  1        Opcode_Entry(1,'DIVB2 ',%X'86'   ,simple_2_operand,context_b),
;  388        M 0517  1        Opcode_Entry(1,'DIVB3 ',%X'87'   ,simple_3_operand,context_b),
;  389        M 0518  1        Opcode_Entry(1,'DIVD2 ',%X'66'   ,simple_2_operand,context_d),
;  390        M 0519  1        Opcode_Entry(1,'DIVD3 ',%X'67'   ,simple_3_operand,context_d),
;  391        M 0520  1        Opcode_Entry(1,'DIVF2 ',%X'46'   ,simple_2_operand,context_f),
;  392        M 0521  1        Opcode_Entry(1,'DIVF3 ',%X'47'   ,simple_3_operand,context_f),
;  393        M 0522  1        Opcode_Entry(1,'DIVG2 ',%X'46FD',simple_2_operand,context_g),
;  394        M 0523  1        Opcode_Entry(1,'DIVG3 ',%X'47FD',simple_3_operand,context_g),
;  395        M 0524  1        Opcode_Entry(1,'DIVH2 ',%X'66FD',simple_2_operand,context_h),
;  396        M 0525  1        Opcode_Entry(1,'DIVH3 ',%X'67FD',simple_3_operand,context_h),
;  397        M 0526  1        Opcode_Entry(1,'DIVL2 ',%X'C6'   ,simple_2_operand,context_l),
;  398        M 0527  1        Opcode_Entry(1,'DIVL3 ',%X'C7'   ,simple_3_operand,context_l),
;  399        M 0528  1        Opcode_Entry(1,'DIVP  ',%X'27'   ,simple_3_operand,context_p),
;  400        M 0529  1        Opcode_Entry(1,'DIVW2 ',%X'A6'   ,simple_2_operand,context_w),
;  401        M 0530  1        Opcode_Entry(1,'DIVW3 ',%X'A7'   ,simple_3_operand,context_w),
;  402        M 0531  1        Opcode_Entry(1,'EDITPC',%X'38'   ,string_4_operand,context_p,context_b),
;  403        M 0532  1        Opcode_Entry(1,'EDIV  ',%X'7B'   ,complex_EDIV),
;  404        M 0533  1        Opcode_Entry(1,'EMODD ',%X'74'   ,complex_EMOD   ,context_bu,context_d),
;  405        M 0534  1        Opcode_Entry(1,'EMODF ',%X'54'   ,complex_EMOD   ,context_bu,context_f),
;  406        M 0535  1        Opcode_Entry(1,'EMODG ',%X'54FD',complex_EMOD   ,context_wu,context_g),!
;  407        M 0536  1        Opcode_Entry(1,'EMODH ',%X'74FD',complex_EMOD   ,context_wu,context_h),!
;  408        M 0537  1        Opcode_Entry(1,'EMUL  ',%X'7A'   ,complex_EMUL),
;  409        M 0538  1        Opcode_Entry(1,'EXTV  ',%X'EE'   ,simple_bit_field,context_m,context_l),
;  410        M 0539  1        Opcode_Entry(1,'EXTZV ',%X'EF'   ,simple_bit_field,context_m,context_l),
;  411        M 0540  1        Opcode_Entry(1,'FFC   ',%X'EB'   ,simple_bit_field,context_m,context_l),
;  412        M 0541  1        Opcode_Entry(1,'FFS   ',%X'EA'   ,simple_bit_field,context_m,context_l),
;  413        M 0542  1        Opcode_Entry(1,'HALT  ',%X'00'   ,simple_0_operand),
;  414        M 0543  1        Opcode_Entry(1,'INCB  ',%X'96'   ,simple_1_operand,context_b),
;  415        M 0544  1        Opcode_Entry(1,'INCL  ',%X'D6'   ,simple_1_operand,context_l),
;  416        M 0545  1        Opcode_Entry(1,'INCW  ',%X'B6'   ,simple_1_operand,context_w),
;  417        M 0546  1        Opcode_Entry(1,'INDEX ',%X'0A'   ,complex_INDEX),
```

```
;  419        M 0547  1        Opcode_Entry(1,'INSQHI',%X'5C'   ,simple_2_operand,context_b),
;  420        M 0548  1        Opcode_Entry(1,'INSQTI',%X'5D'   ,simple_2_operand,context_b),
;  421        M 0549  1        Opcode_Entry(1,'INSQUE',%X'0E'   ,simple_2_operand,context_b),
;  422        M 0550  1        Opcode_Entry(1,'INSV  ',%X'F0'   ,simple_bit_field,context_l,context_m),
;  423        M 0551  1        Opcode_Entry(1,'JMP   ',%X'17'   ,simple_1_operand,context_b),
;  424        M 0552  1        Opcode_Entry(1,'JSB   ',%X'16'   ,simple_1_operand,context_b),
;  425        M 0553  1        Opcode_Entry(1,'LDPCTX',%X'06'   ,simple_0_operand),
;  426        M 0554  1        Opcode_Entry(1,'LOCC  ',%X'3A'   ,locate_character,context_b,context_t),
;  427        M 0555  1        Opcode_Entry(1,'MATCHC',%X'39'   ,simple_2_operand,context_t),
;  428        M 0556  1        Opcode_Entry(1,'MCOMB ',%X'92'   ,simple_2_operand,context_b),
;  429        M 0557  1        Opcode_Entry(1,'MCOML ',%X'D2'   ,simple_2_operand,context_l),
;  430        M 0558  1        Opcode_Entry(1,'MCOMW ',%X'B2'   ,simple_2_operand,context_w),
;  431        M 0559  1        Opcode_Entry(1,'MFPR  ',%X'DB'   ,simple_2_operand,context_l),
;  432        M 0560  1        Opcode_Entry(1,'MNEGB ',%X'8E'   ,simple_2_operand,context_b),
;  433        M 0561  1        Opcode_Entry(1,'MNEGD ',%X'72'   ,simple_2_operand,context_d),
;  434        M 0562  1        Opcode_Entry(1,'MNEGF ',%X'52'   ,simple_2_operand,context_f),
;  435        M 0563  1        Opcode_Entry(1,'MNEGG ',%X'52FD',simple_2_operand,context_g),          !
;  436        M 0564  1        Opcode_Entry(1,'MNEGH ',%X'72FD',simple_2_operand,context_h),          !
;  437        M 0565  1        Opcode_Entry(1,'MNEGL ',%X'CE'   ,simple_2_operand,context_l),
;  438        M 0566  1        Opcode_Entry(1,'MNEGW ',%X'AE'   ,simple_2_operand,context_w),
;  439        M 0567  1        Opcode_Entry(1,'MOVAB ',%X'9E'   ,evaluate_address,context_b,context_l),
;  440        M 0568  1        Opcode_Entry(0,'MOVAD ',%X'7E'   ,evaluate_address,context_d,context_l),
;  441        M 0569  1        Opcode_Entry(0,'MOVAF ',%X'DE'   ,evaluate_address,context_f,context_l),
;  442        M 0570  1        Opcode_Entry(0,'MOVAG ',%X'7E'   ,evaluate_address,context_g,context_l),
;  443        M 0571  1        Opcode_Entry(0,'MOVAH ',%X'7EFD',evaluate_address,context_h,context_l),!
;  444        M 0572  1        Opcode_Entry(1,'MOVAL ',%X'DE'   ,evaluate_address,context_l,context_l),
;  445        M 0573  1        Opcode_Entry(1,'MOVAO ',%X'7EFD',evaluate_address,context_o,context_l),!
;  446        M 0574  1        Opcode_Entry(1,'MOVAQ ',%X'7E'   ,evaluate_address,context_q,context_l),
;  447        M 0575  1        Opcode_Entry(1,'MOVAW ',%X'3E'   ,evaluate_address,context_w,context_l),
;  448        M 0576  1        Opcode_Entry(1,'MOVB  ',%X'90'   ,simple_2_operand,context_b),
;  449        M 0577  1        Opcode_Entry(1,'MOVC3 ',%X'28'   ,string_3_operand,context_t,context_b),
;  450        M 0578  1        Opcode_Entry(1,'MOVC5 ',%X'2C'   ,string_5_operand,context_t,context_t),
;  451        M 0579  1        Opcode_Entry(1,'MOVD  ',%X'70'   ,simple_2_operand,context_d),
;  452        M 0580  1        Opcode_Entry(1,'MOVF  ',%X'50'   ,simple_2_operand,context_f),
;  453        M 0581  1        Opcode_Entry(1,'MOVG  ',%X'50FD',simple_2_operand,context_g),          !
;  454        M 0582  1        Opcode_Entry(1,'MOVH  ',%X'70FD',simple_2_operand,context_h),          !
;  455        M 0583  1        Opcode_Entry(1,'MOVL  ',%X'D0'   ,simple_2_operand,context_l),
;  456        M 0584  1        Opcode_Entry(1,'MOVO  ',%X'7DFD',simple_2_operand,context_o),          !
;  457        M 0585  1        Opcode_Entry(1,'MOVP  ',%X'34'   ,string_3_operand,context_p,context_b),
;  458        M 0586  1        Opcode_Entry(1,'MOVPSL',%X'DC'   ,simple_1_operand,context_l),
;  459        M 0587  1        Opcode_Entry(1,'MOVQ  ',%X'7D'   ,simple_2_operand,context_q),
;  460        M 0588  1        Opcode_Entry(1,'MOVTC ',%X'2E'   ,string_6_operand,context_t,context_t),
;  461        M 0589  1        Opcode_Entry(1,'MOVTUC',%X'2F'   ,string_6_operand,context_t,context_t),
;  462        M 0590  1        Opcode_Entry(1,'MOVW  ',%X'B0'   ,simple_2_operand,context_w),
;  463        M 0591  1        Opcode_Entry(1,'MOVZBL',%X'9A'   ,convert_datatype,context_b,context_l),
;  464        M 0592  1        Opcode_Entry(1,'MOVZBW',%X'9B'   ,convert_datatype,context_b,context_w),
;  465        M 0593  1        Opcode_Entry(1,'MOVZWL',%X'3C'   ,convert_datatype,context_w,context_l),
;  466        M 0594  1        Opcode_Entry(1,'MTPR  ',%X'DA'   ,simple_2_operand,context_l),
```

```
;  468          M 0595  1          Opcode_Entry(1,'MULB2 ',%X'84'  ,simple_2_operand,context_b),
;  469          M 0596  1          Opcode_Entry(1,'MULB3 ',%X'85'  ,simple_3_operand,context_b),
;  470          M 0597  1          Opcode_Entry(1,'MULD2 ',%X'64'  ,simple_2_operand,context_d),
;  471          M 0598  1          Opcode_Entry(1,'MULD3 ',%X'65'  ,simple_3_operand,context_d),
;  472          M 0599  1          Opcode_Entry(1,'MULF2 ',%X'44'  ,simple_2_operand,context_f),
;  473          M 0600  1          Opcode_Entry(1,'MULF3 ',%X'45'  ,simple_3_operand,context_f),
;  474          M 0601  1          Opcode_Entry(1,'MULG2 ',%X'44FD',simple_2_operand,context_g),          !
;  475          M 0602  1          Opcode_Entry(1,'MULG3 ',%X'45FD',simple_3_operand,context_g),          !
;  476          M 0603  1          Opcode_Entry(1,'MULH2 ',%X'64FD',simple_2_operand,context_h),          !
;  477          M 0604  1          Opcode_Entry(1,'MULH3 ',%X'65FD',simple_3_operand,context_h),          !
;  478          M 0605  1          Opcode_Entry(1,'MULL2 ',%X'C4'  ,simple_2_operand,context_l),
;  479          M 0606  1          Opcode_Entry(1,'MULL3 ',%X'C5'  ,simple_3_operand,context_l),
;  480          M 0607  1          Opcode_Entry(1,'MULP  ',%X'25'  ,simple_3_operand,context_p),
;  481          M 0608  1          Opcode_Entry(1,'MULW2 ',%X'A4'  ,simple_2_operand,context_w),
;  482          M 0609  1          Opcode_Entry(1,'MULW3 ',%X'A5'  ,simple_3_operand,context_w),
;  483          M 0610  1          Opcode_Entry(1,'NOP   ',%X'01'  ,simple_0_operand),
;  484          M 0611  1          Opcode_Entry(1,'POLYD ',%X'75'  ,polynomial_value,context_d,context_t),
;  485          M 0612  1          Opcode_Entry(1,'POLYF ',%X'55'  ,polynomial_value,context_f,context_t),
;  486          M 0613  1          Opcode_Entry(1,'POLYG ',%X'55FD',polynomial_value,context_g,context_t),!
;  487          M 0614  1          Opcode_Entry(1,'POLYH ',%X'75FD',polynomial_value,context_h,context_t),!
;  488          M 0615  1          Opcode_Entry(1,'POPR  ',%X'BA'  ,simple_1_operand,context_w),
;  489          M 0616  1          Opcode_Entry(1,'PROBER',%X'0C'  ,probe_for_access,context_b,context_t),
;  490          M 0617  1          Opcode_Entry(1,'PROBEW',%X'0D'  ,probe_for_access,context_b,context_t),
;  491          M 0618  1          Opcode_Entry(1,'PUSHAB',%X'9F'  ,simple_1_operand,context_b),
;  492          M 0619  1          Opcode_Entry(0,'PUSHAD',%X'7F'  ,simple_1_operand,context_d),
;  493          M 0620  1          Opcode_Entry(0,'PUSHAF',%X'DF'  ,simple_1_operand,context_f),
;  494          M 0621  1          Opcode_Entry(0,'PUSHAG',%X'7F'  ,simple_1_operand,context_g),
;  495          M 0622  1          Opcode_Entry(0,'PUSHAH',%X'7FFD',simple_1_operand,context_h),          !
;  496          M 0623  1          Opcode_Entry(1,'PUSHAL',%X'DF'  ,simple_1_operand,context_l),
;  497          M 0624  1          Opcode_Entry(1,'PUSHAO',%X'7FFD',simple_1_operand,context_o),          !
;  498          M 0625  1          Opcode_Entry(1,'PUSHAQ',%X'7F'  ,simple_1_operand,context_q),
;  499          M 0626  1          Opcode_Entry(1,'PUSHAW',%X'3F'  ,simple_1_operand,context_w),
;  500          M 0627  1          Opcode_Entry(1,'PUSHL ',%X'DD'  ,simple_1_operand,context_l),
;  501          M 0628  1          Opcode_Entry(1,'PUSHR ',%X'BB'  ,simple_1_operand,context_w),
;  502          M 0629  1          Opcode_Entry(1,'REI   ',%X'02'  ,simple_0_operand),
;  503          M 0630  1          Opcode_Entry(1,'REMQHI',%X'5E'  ,simple_2_operand,context_b),
;  504          M 0631  1          Opcode_Entry(1,'REMQTI',%X'5F'  ,simple_2_operand,context_b),
;  505          M 0632  1          Opcode_Entry(1,'REMQUE',%X'0F'  ,simple_2_operand,context_b),
;  506          M 0633  1          Opcode_Entry(1,'RET   ',%X'04'  ,simple_0_operand),
;  507          M 0634  1          Opcode_Entry(1,'ROTL  ',%X'9C'  ,complex_SHIFT   ,context_l),
;  508          M 0635  1          Opcode_Entry(1,'RSB   ',%X'05'  ,simple_0_operand),
;  509          M 0636  1          Opcode_Entry(1,'SBWC  ',%X'D9'  ,simple_2_operand,context_l),
;  510          M 0637  1          Opcode_Entry(1,'SCANC ',%X'2A'  ,string_4_operand,context_t,context_b),
;  511          M 0638  1          Opcode_Entry(1,'SKPC  ',%X'3B'  ,locate_character,context_b,context_t),
;  512          M 0639  1          Opcode_Entry(1,'SOBGEQ',%X'F4'  ,branch_1_operand,context_l,context_b),
;  513          M 0640  1          Opcode_Entry(1,'SOBGTR',%X'F5'  ,branch_1_operand,context_l,context_b),
;  514          M 0641  1          Opcode_Entry(1,'SPANC ',%X'2B'  ,string_4_operand,context_t,context_b),
```

```
;  516          M 0642  1          Opcode_Entry(1,'SUBB2 ',%X'82'  ,simple_2_operand,context_b),
;  517          M 0643  1          Opcode_Entry(1,'SUBB3 ',%X'83'  ,simple_3_operand,context_b),
;  518          M 0644  1          Opcode_Entry(1,'SUBD2 ',%X'62'  ,simple_2_operand,context_d),
;  519          M 0645  1          Opcode_Entry(1,'SUBD3 ',%X'63'  ,simple_3_operand,context_d),
;  520          M 0646  1          Opcode_Entry(1,'SUBF2 ',%X'42'  ,simple_2_operand,context_f),
;  521          M 0647  1          Opcode_Entry(1,'SUBF3 ',%X'43'  ,simple_3_operand,context_f),
;  522          M 0648  1          Opcode_Entry(1,'SUBG2 ',%X'42FD',simple_2_operand,context_g),      !
;  523          M 0649  1          Opcode_Entry(1,'SUBG3 ',%X'43FD',simple_3_operand,context_g),      !
;  524          M 0650  1          Opcode_Entry(1,'SUBH2 ',%X'62FD',simple_2_operand,context_h),      !
;  525          M 0651  1          Opcode_Entry(1,'SUBH3 ',%X'63FD',simple_3_operand,context_h),      !
;  526          M 0652  1          Opcode_Entry(1,'SUBL2 ',%X'C2'  ,simple_2_operand,context_l),
;  527          M 0653  1          Opcode_Entry(1,'SUBL3 ',%X'C3'  ,simple_3_operand,context_l),
;  528          M 0654  1          Opcode_Entry(1,'SUBP4 ',%X'22'  ,simple_2_operand,context_p),
;  529          M 0655  1          Opcode_Entry(1,'SUBP6 ',%X'23'  ,simple_3_operand,context_p),
;  530          M 0656  1          Opcode_Entry(1,'SUBW2 ',%X'A2'  ,simple_2_operand,context_w),
;  531          M 0657  1          Opcode_Entry(1,'SUBW3 ',%X'A3'  ,simple_3_operand,context_w),
;  532          M 0658  1          Opcode_Entry(1,'SVPCTX',%X'07'  ,simple_0_operand),
;  533          M 0659  1          Opcode_Entry(1,'TSTB  ',%X'95'  ,simple_1_operand,context_b),
;  534          M 0660  1          Opcode_Entry(1,'TSTD  ',%X'73'  ,simple_1_operand,context_d),
;  535          M 0661  1          Opcode_Entry(1,'TSTF  ',%X'53'  ,simple_1_operand,context_f),
;  536          M 0662  1          Opcode_Entry(1,'TSTG  ',%X'53FD',simple_1_operand,context_g),      !
;  537          M 0663  1          Opcode_Entry(1,'TSTH  ',%X'73FD',simple_1_operand,context_h),      !
;  538          M 0664  1          Opcode_Entry(1,'TSTL  ',%X'D5'  ,simple_1_operand,context_l),
;  539          M 0665  1          Opcode_Entry(1,'TSTW  ',%X'B5'  ,simple_1_operand,context_w),
;  540          M 0666  1          Opcode_Entry(1,'XFC   ',%X'FC'  ,simple_0_operand),                !
;  541          M 0667  1          Opcode_Entry(1,'XORB2 ',%X'8C'  ,simple_2_operand,context_b),
;  542          M 0668  1          Opcode_Entry(1,'XORB3 ',%X'8D'  ,simple_3_operand,context_b),
;  543          M 0669  1          Opcode_Entry(1,'XORL2 ',%X'CC'  ,simple_2_operand,context_l),
;  544          M 0670  1          Opcode_Entry(1,'XORL3 ',%X'CD'  ,simple_3_operand,context_l),
;  545          M 0671  1          Opcode_Entry(1,'XORW2 ',%X'AC'  ,simple_2_operand,context_w),
;  546          M 0672  1          Opcode_Entry(1,'XORW3 ',%X'AD'  ,simple_3_operand,context_w)
;  547          M 0673  1
;  548            0674  1 %;                  ! ****   End of definition of MACRO Opcode_List
```

```
;  550          0675  1 !++
;  551          0676  1 !
;  552          0677  1 !       Used in the macro Opcode_List which is used in the following
;  553          0678  1 !       GLOBAL BIND to define DBG$Opcode_Name_table
;  554          0679  1 !
;  555          0680  1 !--
;  556        M 0681  1 MACRO Opcode_Entry(Flag, Name, Code, Kind, Arg1, Arg2) =
;  557        M 0682  1         BYTE(%ASCII Name),
;  558        M 0683  1         WORD(%IF ((Code) GTRU %X'FF') %THEN (Code) %ELSE (Code)^8 %FI),
;  559        M 0684  1         BYTE(Kind),
;  560        M 0685  1         BYTE(%IF (%LENGTH LSS 5) %THEN 0 %ELSE
;  561        M 0686  1              %IF (%LENGTH EQL 5) %THEN (Arg1) %ELSE (((Arg1)^4)+(Arg2))
;  562          0687  1              %FI %FI) %;
;  563          0688  1
;  564          0689  1 GLOBAL BIND
;  565          0690  1     DBG$Opcode_Name_Table =
;  566          0691  1         UPLIT WORD(BYTE(%ASCII '??????'),WORD(0,0),Opcode_List) : BLOCKVECTOR [,10,BYTE];
;  567          0692  1
;  568          0693  1 !++
;  569          0694  1 !
;  570          0695  1 !       Undeclare Opcode_Entry so that another definition of the macro
;  571          0696  1 !       Opcode_Entry can be defined the to build the back translation
;  572          0697  1 !       table, DBG$Opcode_Kind_Table.
;  573          0698  1 !--
;  574          0699  1 UNDECLARE %QUOTE Opcode_Entry;
;  575          0700  1
;  576        M 0701  1 MACRO Opcode_Entry(Flag, Name, Code, Kind, Arg1, Arg2) =
;  577        M 0702  1 %ASSIGN(Opcode_Index, Opcode_Index + 1)
;  578        M 0703  1 %IF Flag
;  579        M 0704  1 %THEN
;  580        M 0705  1   [IF (Code LSS %X'100') THEN (Code) ELSE (((Code)^-8)+%X'100')] = Opcode_Index
;  581        M 0706  1 %ELSE
;  582        M 0707  1 %ASSIGN(Ignore_Index, Ignore_Index + 1)
;  583        M 0708  1   [255 + Ignore_Index] = 0
;  584        M 0709  1 %FI
;  585          0710  1  %;
;  586          0711  1
;  587          0712  1 COMPILETIME
;  588          0713  1     Opcode_Index = 0,
;  589          0714  1     Ignore_Index = 0;
;  590          0715  1
;  591          0716  1 GLOBAL
;  592          0717  1     DBG$Opcode_Kind_Table : PSECT(DBG$PLIT) VECTOR[512,WORD] PRESET(Opcode_List);
;  593          0718  1
;  594          0719  1 UNDECLARE %QUOTE Opcode_Entry;
;  595          0720  1
;  596          0721  1 LITERAL Opcode_Table_Size = Opcode_Index;
```

```
;  598          0722  1 OWN
;  599          0723  1     Op_Buffer           : BLOCK [16,BYTE],
;  600          0724  1     Data_Size           : PSECT(DBG$PLIT) VECTOR[12,BYTE] PRESET(
;  601          0725  1         [context_b] =  1,               [context_w] =  2,
;  602          0726  1         [context_l] =  4,               [context_q] =  8,
;  603          0727  1         [context_o] = 16,
;  604          0728  1         [context_f] =  4,               [context_d] =  8,
;  605          0729  1         [context_g] =  8,               [context_h] = 16,
;  606          0730  1         [context_bu] = 1,               [context_wu] = 2),
;  607          0731  1     Data_Type           : PSECT(DBG$PLIT) VECTOR[12,BYTE] PRESET(
;  608          0732  1         [context_b] = dsc$k_dtype_b,    [context_w] = dsc$k_dtype_w,
;  609          0733  1         [context_l] = dsc$k_dtype_l,    [context_q] = dsc$k_dtype_q,
;  610          0734  1         [context_o] = dsc$k_dtype_o,
;  611          0735  1         [context_f] = dsc$k_dtype_f,    [context_d] = dsc$k_dtype_d,
;  612          0736  1         [context_g] = dsc$k_dtype_g,    [context_h] = dsc$k_dtype_h,
;  613          0737  1         [context_bu]= dsc$k_dtype_bu,   [context_wu]= dsc$k_dtype_wu);
;  614          0738  1
;  615          0739  1 BIND
;  616          0740  1     Operand_Value = Op_Buffer : LONG,
;  617          0741  1     Register_Name = UPLIT BYTE(
;  618          0742  1         2,'R','0', 0 ,   2,'R','1', 0 ,   2,'R','2', 0 ,   2,'R','3', 0 ,
;  619          0743  1         2,'R','4', 0 ,   2,'R','5', 0 ,   2,'R','6', 0 ,   2,'R','7', 0 ,
;  620          0744  1         2,'R','8', 0 ,   2,'R','9', 0 ,   3,'R','1','0',   3,'R','1','1',
;  621          0745  1         2,'A','P', 0 ,   2,'F','P', 0 ,   2,'S','P', 0 ,   2,'P','C', 0 ,
;  622          0746  1 !
;  623          0747  1         2,'?','?', 0 ,   2,'?','?', 0 ,   2,'I','V', 0 ,   2,'D','V', 0 )
;  624          0748  1                                                 : VECTOR[,LONG],
;  625          0749  1     Format_AD = UPLIT BYTE(%ASCIC '!AD'),
;  626          0750  1     Format_AC = UPLIT BYTE(%ASCIC '!AC'),
;  627          0751  1     comma     = UPLIT BYTE(',');
;  628          0752  1
;  629          0753  1 MACRO
;  630          0754  1     offset = 0,0, 0,0 %,
;  631          0755  1     u_byte = 0,0, 8,0 %,
;  632          0756  1     u_word = 0,0,16,0 %,
;  633          0757  1     u_long = 0,0,32,0 %,
;  634          0758  1     s_byte = 0,0, 8,1 %,
;  635          0759  1     s_word = 0,0,16,1 %,
;  636          0760  1     s_long = 0,0,32,1 %;
;  637          0761  1
;  638          0762  1 FIELD Opcode_Entry_Fields =
;  639          0763  1     SET
;  640          0764  1     op_name      = [0,0,0,0],
;  641          0765  1     op_code      = [6,0,16,0],                          ! Whole opcode
;  642          0766  1     op_code_one  = [6,0,8,0],
;  643          0767  1     op_code_two  = [7,0,8,0],
;  644          0768  1     op_kind      = [8,0,8,0],
;  645          0769  1     op_type      = [9,0,8,0],
;  646          0770  1     op_type_one  = [9,0,4,0],
;  647          0771  1     op_type_two  = [9,4,4,0]
;  648          0772  1     TES;
;  649          0773  1
;  650          0774  1 FIELD Encode_Fields =
;  651          0775  1     SET
;  652          0776  1     Enc_Input_Desc    = [ 0, 0, 0,0],
;  653          0777  1     Enc_Input_Length  = [ 0, 0,16,0],
;  654          0778  1     Enc_Input_Dtype   = [ 2, 0, 8,0],
```

```
;  655          0779  1      Enc_Input_Class   = [ 3, 0, 8,0],
;  656          0780  1      Enc_Input_Buffer  = [ 4, 0,32,0],
;  657          0781  1      Enc_Output_Length = [ 8, 0,32,0],
;  658          0782  1      Enc_Output_Buffer = [12, 0,32,0],
;  659          0783  1      Enc_Final_Address = [16, 0,32,0]
;  660          0784  1      TES;
```

```
;  662          0785  1 GLOBAL ROUTINE DBG$Ins_Decode(Start_Address,Print_Flag,Entry_Flag) =
;  663          0786  2     BEGIN
;  664          0787  2     BUILTIN ACTUALCOUNT;
;  665          0788  2     LOCAL Pointer;
;  666          0789  2     LOCAL
;  667          0790  2         Signal_Flag     : VOLATILE,
;  668          0791  2         Error_Value     : VOLATILE;
;  669          0792  2
;  670          0793  2     ROUTINE Decode_Handler(Sig_Args,Mech_Args,Enable_Args) =
;  671          0794  3         BEGIN
;  672          0795  3         MAP Sig_Args    : REF BLOCK[,BYTE],
;  673          0796  3             Mech_Args   : REF BLOCK[,BYTE],
;  674          0797  3             Enable_Args : REF VECTOR[,LONG];
;  675          0798  3         EXTERNAL ROUTINE SYS$UNWIND : Addressing_Mode(General);
;  676          0799  3
;  677          0800  3         IF .Sig_Args[chf$l_sig_name] EQL ss$_unwind THEN RETURN ss$_continue;
;  678          0801  3         IF .(.Enable_Args[1]) THEN RETURN ss$_resignal;
;  679          0802  3
;  680          0803  3         Mech_Args[chf$l_mch_savr0] = .(.Enable_Args[2]);
;  681          0804  3
;  682          0805  3         SYS$UNWIND(0,0);
;  683          0806  3         RETURN ss$_continue;
;  684          0807  2         END;


                                                        .TITLE  DBGENCDEC
                                                        .IDENT  \V04-000\

                                                        .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

3F  3F  3F  3F  3F  3F  00000 P.AAA:    .ASCII  \??????\
                0000  0000  00006       .WORD   0, 0
20  20  42  42  43  41  0000A          .ASCII  \ACBB  \
                9D00        00010       .WORD   -25344
                  07        00012       .BYTE   7
                  01        00013       .BYTE   1
20  20  44  42  43  41  00014          .ASCII  \ACBD  \
                6F00        0001A       .WORD   28416
                  07        0001C       .BYTE   7
                  61        0001D       .BYTE   97
20  20  46  42  43  41  0001E          .ASCII  \ACBF  \
                4F00        00024       .WORD   20224
                  07        00026       .BYTE   7
                  51        00027       .BYTE   81
20  20  47  42  43  41  00028          .ASCII  \ACBG  \
                4FFD        0002E       .WORD   20477
                  07        00030       .BYTE   7
                  71        00031       .BYTE   113
20  20  48  42  43  41  00032          .ASCII  \ACBH  \
                6FFD        00038       .WORD   28669
                  07        0003A       .BYTE   7
                  81        0003B       .BYTE   -127
20  20  4C  42  43  41  0003C          .ASCII  \ACBL  \
                F100        00042       .WORD   -3840
                  07        00044       .BYTE   7
                  21        00045       .BYTE   33
20  20  57  42  43  41  00046          .ASCII  \ACBW  \
```

```
                                3D00  0004C          .WORD   15616
                                  07  0004E          .BYTE   7
                                  11  0004F          .BYTE   17
                20  49  57  41  44  41  00050          .ASCII  \ADAWI \
                                5800  00056          .WORD   22528
                                  02  00058          .BYTE   2
                                  01  00059          .BYTE   1
                20  32  42  44  44  41  0005A          .ASCII  \ADDB2 \
                                8000  00060          .WORD   -32768
                                  02  00062          .BYTE   2
                                  00  00063          .BYTE   0
                20  33  42  44  44  41  00064          .ASCII  \ADDB3 \
                                8100  0006A          .WORD   -32512
                                  03  0006C          .BYTE   3
                                  00  0006D          .BYTE   0
                20  32  44  44  44  41  0006E          .ASCII  \ADDD2 \
                                6000  00074          .WORD   24576
                                  02  00076          .BYTE   2
                                  06  00077          .BYTE   6
                20  33  44  44  44  41  00078          .ASCII  \ADDD3 \
                                6100  0007E          .WORD   24832
                                  03  00080          .BYTE   3
                                  06  00081          .BYTE   6
                20  32  46  44  44  41  00082          .ASCII  \ADDF2 \
                                4000  00088          .WORD   16384
                                  02  0008A          .BYTE   2
                                  05  0008B          .BYTE   5
                20  33  46  44  44  41  0008C          .ASCII  \ADDF3 \
                                4100  00092          .WORD   16640
                                  03  00094          .BYTE   3
                                  05  00095          .BYTE   5
                20  32  47  44  44  41  00096          .ASCII  \ADDG2 \
                                40FD  0009C          .WORD   16637
                                  02  0009E          .BYTE   2
                                  07  0009F          .BYTE   7
                20  33  47  44  44  41  000A0          .ASCII  \ADDG3 \
                                41FD  000A6          .WORD   16893
                                  03  000A8          .BYTE   3
                                  07  000A9          .BYTE   7
                20  32  48  44  44  41  000AA          .ASCII  \ADDH2 \
                                60FD  000B0          .WORD   24829
                                  02  000B2          .BYTE   2
                                  08  000B3          .BYTE   8
                20  33  48  44  44  41  000B4          .ASCII  \ADDH3 \
                                61FD  000BA          .WORD   25085
                                  03  000BC          .BYTE   3
                                  08  000BD          .BYTE   8
                20  32  4C  44  44  41  000BE          .ASCII  \ADDL2 \
                                C000  000C4          .WORD   -16384
                                  02  000C6          .BYTE   2
                                  02  000C7          .BYTE   2
                20  33  4C  44  44  41  000C8          .ASCII  \ADDL3 \
                                C100  000CE          .WORD   -16128
                                  03  000D0          .BYTE   3
                                  02  000D1          .BYTE   2
                20  34  50  44  44  41  000D2          .ASCII  \ADDP4 \
                                2000  000D8          .WORD   8192
```

```
                                    02  000DA          .BYTE   2
                                    0C  000DB          .BYTE   12
                20  36  50  44  44  41  000DC          .ASCII  \ADDP6 \
                                  2100  000E2          .WORD   8448
                                    03  000E4          .BYTE   3
                                    0C  000E5          .BYTE   12
                20  32  57  44  44  41  000E6          .ASCII  \ADDW2 \
                                  A000  000EC          .WORD   -24576
                                    02  000EE          .BYTE   2
                                    01  000EF          .BYTE   1
                20  33  57  44  44  41  000F0          .ASCII  \ADDW3 \
                                  A100  000F6          .WORD   -24320
                                    03  000F8          .BYTE   3
                                    01  000F9          .BYTE   1
                20  20  43  57  44  41  000FA          .ASCII  \ADWC  \
                                  D800  00100          .WORD   -10240
                                    02  00102          .BYTE   2
                                    02  00103          .BYTE   2
                51  45  4C  42  4F  41  00104          .ASCII  \AOBLEQ\
                                  F300  0010A          .WORD   -3328
                                    06  0010C          .BYTE   6
                                    20  0010D          .BYTE   32
                53  53  4C  42  4F  41  0010E          .ASCII  \AOBLSS\
                                  F200  00114          .WORD   -3584
                                    06  00116          .BYTE   6
                                    20  00117          .BYTE   32
                20  20  4C  48  53  41  00118          .ASCII  \ASHL  \
                                  7800  0011E          .WORD   30720
                                    10  00120          .BYTE   16
                                    02  00121          .BYTE   2
                20  20  50  48  53  41  00122          .ASCII  \ASHP  \
                                  F800  00128          .WORD   -2048
                                    17  0012A          .BYTE   23
                                    0C  0012B          .BYTE   12
                20  20  51  48  53  41  0012C          .ASCII  \ASHQ  \
                                  7900  00132          .WORD   30976
                                    10  00134          .BYTE   16
                                    03  00135          .BYTE   3
                20  20  20  43  42  42  00136          .ASCII  \BBC   \
                                  E100  0013C          .WORD   -7936
                                    05  0013E          .BYTE   5
                                    E0  0013F          .BYTE   -32
                20  20  43  43  42  42  00140          .ASCII  \BBCC  \
                                  E500  00146          .WORD   -6912
                                    05  00148          .BYTE   5
                                    E0  00149          .BYTE   -32
                20  49  43  43  42  42  0014A          .ASCII  \BBCCI \
                                  E700  00150          .WORD   -6400
                                    05  00152          .BYTE   5
                                    E0  00153          .BYTE   -32
                20  20  53  43  42  42  00154          .ASCII  \BBCS  \
                                  E300  0015A          .WORD   -7424
                                    05  0015C          .BYTE   5
                                    E0  0015D          .BYTE   -32
                20  20  20  53  42  42  0015E          .ASCII  \BBS   \
                                  E000  00164          .WORD   -8192
                                    05  00166          .BYTE   5
```

```
                                    E0  00167          .BYTE   -32
20  20  43  53  42  42             00168          .ASCII  \BBSC  \
                              E400  0016E          .WORD   -7168
                                 05 00170          .BYTE   5
                                    E0  00171          .BYTE   -32
20  20  53  53  42  42             00172          .ASCII  \BBSS  \
                              E200  00178          .WORD   -7680
                                 05 0017A          .BYTE   5
                                    E0  0017B          .BYTE   -32
20  49  53  53  42  42             0017C          .ASCII  \BBSSI \
                              E600  00182          .WORD   -6656
                                 05 00184          .BYTE   5
                                    E0  00185          .BYTE   -32
20  20  20  43  43  42             00186          .ASCII  \BCC   \
                              1E00  0018C          .WORD   7680
                                 04 0018E          .BYTE   4
                                 00 0018F          .BYTE   0
20  20  20  53  43  42             00190          .ASCII  \BCS   \
                              1F00  00196          .WORD   7936
                                 04 00198          .BYTE   4
                                 00 00199          .BYTE   0
20  20  4C  51  45  42             0019A          .ASCII  \BEQL  \
                              1300  001A0          .WORD   4864
                                 04 001A2          .BYTE   4
                                 00 001A3          .BYTE   0
20  55  4C  51  45  42             001A4          .ASCII  \BEQLU \
                              1300  001AA          .WORD   4864
                                 04 001AC          .BYTE   4
                                 00 001AD          .BYTE   0
20  20  51  45  47  42             001AE          .ASCII  \BGEQ  \
                              1800  001B4          .WORD   6144
                                 04 001B6          .BYTE   4
                                 00 001B7          .BYTE   0
20  55  51  45  47  42             001B8          .ASCII  \BGEQU \
                              1E00  001BE          .WORD   7680
                                 04 001C0          .BYTE   4
                                 00 001C1          .BYTE   0
20  20  52  54  47  42             001C2          .ASCII  \BGTR  \
                              1400  001C8          .WORD   5120
                                 04 001CA          .BYTE   4
                                 00 001CB          .BYTE   0
20  55  52  54  47  42             001CC          .ASCII  \BGTRU \
                              1A00  001D2          .WORD   6656
                                 04 001D4          .BYTE   4
                                 00 001D5          .BYTE   0
20  32  42  43  49  42             001D6          .ASCII  \BICB2 \
                              8A00  001DC          .WORD   -30208
                                 02 001DE          .BYTE   2
                                 00 001DF          .BYTE   0
20  33  42  43  49  42             001E0          .ASCII  \BICB3 \
                              8B00  001E6          .WORD   -29952
                                 03 001E8          .BYTE   3
                                 00 001E9          .BYTE   0
20  32  4C  43  49  42             001EA          .ASCII  \BICL2 \
                              CA00  001F0          .WORD   -13824
                                 02 001F2          .BYTE   2
                                 02 001F3          .BYTE   2
```

```
                20  33  4C  43  49  42  001F4          .ASCII  \BICL3 \
                                CB00    001FA          .WORD   -13568
                                  03    001FC          .BYTE   3
                                  02    001FD          .BYTE   2
                57  53  50  43  49  42  001FE          .ASCII  \BICPSW\
                                B900    00204          .WORD   -18176
                                  01    00206          .BYTE   1
                                  01    00207          .BYTE   1
                20  32  57  43  49  42  00208          .ASCII  \BICW2 \
                                AA00    0020E          .WORD   -22016
                                  02    00210          .BYTE   2
                                  01    00211          .BYTE   1
                20  33  57  43  49  42  00212          .ASCII  \BICW3 \
                                AB00    00218          .WORD   -21760
                                  03    0021A          .BYTE   3
                                  01    0021B          .BYTE   1
                20  32  42  53  49  42  0021C          .ASCII  \BISB2 \
                                8800    00222          .WORD   -30720
                                  02    00224          .BYTE   2
                                  00    00225          .BYTE   0
                20  33  42  53  49  42  00226          .ASCII  \BISB3 \
                                8900    0022C          .WORD   -30464
                                  03    0022E          .BYTE   3
                                  00    0022F          .BYTE   0
                20  32  4C  53  49  42  00230          .ASCII  \BISL2 \
                                C800    00236          .WORD   -14336
                                  02    00238          .BYTE   2
                                  02    00239          .BYTE   2
                20  33  4C  53  49  42  0023A          .ASCII  \BISL3 \
                                C900    00240          .WORD   -14080
                                  03    00242          .BYTE   3
                                  02    00243          .BYTE   2
                57  53  50  53  49  42  00244          .ASCII  \BISPSW\
                                B800    0024A          .WORD   -18432
                                  01    0024C          .BYTE   1
                                  01    0024D          .BYTE   1
                20  32  57  53  49  42  0024E          .ASCII  \BISW2 \
                                A800    00254          .WORD   -22528
                                  02    00256          .BYTE   2
                                  01    00257          .BYTE   1
                20  33  57  53  49  42  00258          .ASCII  \BISW3 \
                                A900    0025E          .WORD   -22272
                                  03    00260          .BYTE   3
                                  01    00261          .BYTE   1
                20  20  42  54  49  42  00262          .ASCII  \BITB  \
                                9300    00268          .WORD   -27904
                                  02    0026A          .BYTE   2
                                  00    0026B          .BYTE   0
                20  20  4C  54  49  42  0026C          .ASCII  \BITL  \
                                D300    00272          .WORD   -11520
                                  02    00274          .BYTE   2
                                  02    00275          .BYTE   2
                20  20  57  54  49  42  00276          .ASCII  \BITW  \
                                B300    0027C          .WORD   -19712
                                  02    0027E          .BYTE   2
                                  01    0027F          .BYTE   1
                20  20  43  42  4C  42  00280          .ASCII  \BLBC  \
```

```
                                E900  00286          .WORD   -5888
                                  05  00288          .BYTE   5
                                  20  00289          .BYTE   32
20  20  53  42  4C  42           0028A          .ASCII  \BLBS  \
                                E800  00290          .WORD   -6144
                                  05  00292          .BYTE   5
                                  20  00293          .BYTE   32
20  20  51  45  4C  42           00294          .ASCII  \BLEQ  \
                                1500  0029A          .WORD   5376
                                  04  0029C          .BYTE   4
                                  00  0029D          .BYTE   0
20  55  51  45  4C  42           0029E          .ASCII  \BLEQU \
                                1B00  002A4          .WORD   6912
                                  04  002A6          .BYTE   4
                                  00  002A7          .BYTE   0
20  20  53  53  4C  42           002A8          .ASCII  \BLSS  \
                                1900  002AE          .WORD   6400
                                  04  002B0          .BYTE   4
                                  00  002B1          .BYTE   0
20  55  53  53  4C  42           002B2          .ASCII  \BLSSU \
                                1F00  002B8          .WORD   7936
                                  04  002BA          .BYTE   4
                                  00  002BB          .BYTE   0
20  20  51  45  4E  42           002BC          .ASCII  \BNEQ  \
                                1200  002C2          .WORD   4608
                                  04  002C4          .BYTE   4
                                  00  002C5          .BYTE   0
20  55  51  45  4E  42           002C6          .ASCII  \BNEQU \
                                1200  002CC          .WORD   4608
                                  04  002CE          .BYTE   4
                                  00  002CF          .BYTE   0
20  20  20  54  50  42           002D0          .ASCII  \BPT   \
                                0300  002D6          .WORD   768
                                  00  002D8          .BYTE   0
                                  00  002D9          .BYTE   0
20  20  20  42  52  42           002DA          .ASCII  \BRB   \
                                1100  002E0          .WORD   4352
                                  04  002E2          .BYTE   4
                                  00  002E3          .BYTE   0
20  20  20  57  52  42           002E4          .ASCII  \BRW   \
                                3100  002EA          .WORD   12544
                                  04  002EC          .BYTE   4
                                  01  002ED          .BYTE   1
20  20  42  42  53  42           002EE          .ASCII  \BSBB  \
                                1000  002F4          .WORD   4096
                                  04  002F6          .BYTE   4
                                  00  002F7          .BYTE   0
20  20  57  42  53  42           002F8          .ASCII  \BSBW  \
                                3000  002FE          .WORD   12288
                                  04  00300          .BYTE   4
                                  01  00301          .BYTE   1
20  20  4C  47  55  42           00302          .ASCII  \BUGL  \
                                FDFF  00308          .WORD   -513
                                  00  0030A          .BYTE   0
                                  00  0030B          .BYTE   0
20  20  57  47  55  42           0030C          .ASCII  \BUGW  \
                                FEFF  00312          .WORD   -257
```

```
                                    00  00314          .BYTE   0
                                    00  00315          .BYTE   0
                20  20  20  43  56  42  00316          .ASCII  \BVC    \
                                  1C00  0031C          .WORD   7168
                                    04  0031E          .BYTE   4
                                    00  0031F          .BYTE   0
                20  20  20  53  56  42  00320          .ASCII  \BVS    \
                                  1D00  00326          .WORD   7424
                                    04  00328          .BYTE   4
                                    00  00329          .BYTE   0
                20  47  4C  4C  41  43  0032A          .ASCII  \CALLG  \
                                  FA00  00330          .WORD   -1536
                                    08  00332          .BYTE   8
                                    00  00333          .BYTE   0
                20  53  4C  4C  41  43  00334          .ASCII  \CALLS  \
                                  FB00  0033A          .WORD   -1280
                                    08  0033C          .BYTE   8
                                    20  0033D          .BYTE   32
                20  42  45  53  41  43  0033E          .ASCII  \CASEB  \
                                  8F00  00344          .WORD   -28928
                                    11  00346          .BYTE   17
                                    00  00347          .BYTE   0
                20  4C  45  53  41  43  00348          .ASCII  \CASEL  \
                                  CF00  0034E          .WORD   -12544
                                    11  00350          .BYTE   17
                                    02  00351          .BYTE   2
                20  57  45  53  41  43  00352          .ASCII  \CASEW  \
                                  AF00  00358          .WORD   -20736
                                    11  0035A          .BYTE   17
                                    01  0035B          .BYTE   1
                20  20  45  4D  48  43  0035C          .ASCII  \CHME   \
                                  BD00  00362          .WORD   -17152
                                    01  00364          .BYTE   1
                                    01  00365          .BYTE   1
                20  20  4B  4D  48  43  00366          .ASCII  \CHMK   \
                                  BC00  0036C          .WORD   -17408
                                    01  0036E          .BYTE   1
                                    01  0036F          .BYTE   1
                20  20  53  4D  48  43  00370          .ASCII  \CHMS   \
                                  BE00  00376          .WORD   -16896
                                    01  00378          .BYTE   1
                                    01  00379          .BYTE   1
                20  20  55  4D  48  43  0037A          .ASCII  \CHMU   \
                                  BF00  00380          .WORD   -16640
                                    01  00382          .BYTE   1
                                    01  00383          .BYTE   1
                20  20  42  52  4C  43  00384          .ASCII  \CLRB   \
                                  9400  0038A          .WORD   -27648
                                    01  0038C          .BYTE   1
                                    00  0038D          .BYTE   0
                20  20  44  52  4C  43  0038E          .ASCII  \CLRD   \
                                  7C00  00394          .WORD   31744
                                    01  00396          .BYTE   1
                                    06  00397          .BYTE   6
                20  20  46  52  4C  43  00398          .ASCII  \CLRF   \
                                  D400  0039E          .WORD   -11264
                                    01  003A0          .BYTE   1
```

```
                                    05  003A1        .BYTE   5
20  20  47  52  4C  43             43  003A2        .ASCII  \CLRG  \
                                7C00  003A8        .WORD   31744
                                    01  003AA        .BYTE   1
                                    07  003AB        .BYTE   7
20  20  48  52  4C  43             43  003AC        .ASCII  \CLRH  \
                                7CFD  003B2        .WORD   31997
                                    01  003B4        .BYTE   1
                                    08  003B5        .BYTE   8
20  20  4C  52  4C  43             43  003B6        .ASCII  \CLRL  \
                                D400  003BC        .WORD   -11264
                                    01  003BE        .BYTE   1
                                    02  003BF        .BYTE   2
20  20  4F  52  4C  43             43  003C0        .ASCII  \CLRO  \
                                7CFD  003C6        .WORD   31997
                                    01  003C8        .BYTE   1
                                    04  003C9        .BYTE   4
20  20  51  52  4C  43             43  003CA        .ASCII  \CLRQ  \
                                7C00  003D0        .WORD   31744
                                    01  003D2        .BYTE   1
                                    03  003D3        .BYTE   3
20  20  57  52  4C  43             43  003D4        .ASCII  \CLRW  \
                                B400  003DA        .WORD   -19456
                                    01  003DC        .BYTE   1
                                    01  003DD        .BYTE   1
20  20  42  50  4D  43             43  003DE        .ASCII  \CMPB  \
                                9100  003E4        .WORD   -28416
                                    02  003E6        .BYTE   2
                                    00  003E7        .BYTE   0
20  33  43  50  4D  43             43  003E8        .ASCII  \CMPC3 \
                                2900  003EE        .WORD   10496
                                    0A  003F0        .BYTE   10
                                    B0  003F1        .BYTE   -80
20  35  43  50  4D  43             43  003F2        .ASCII  \CMPC5 \
                                2D00  003F8        .WORD   11520
                                    0C  003FA        .BYTE   12
                                    BB  003FB        .BYTE   -69
20  20  44  50  4D  43             43  003FC        .ASCII  \CMPD  \
                                7100  00402        .WORD   28928
                                    02  00404        .BYTE   2
                                    06  00405        .BYTE   6
20  20  46  50  4D  43             43  00406        .ASCII  \CMPF  \
                                5100  0040C        .WORD   20736
                                    02  0040E        .BYTE   2
                                    05  0040F        .BYTE   5
20  20  47  50  4D  43             43  00410        .ASCII  \CMPG  \
                                51FD  00416        .WORD   20989
                                    02  00418        .BYTE   2
                                    07  00419        .BYTE   7
20  20  48  50  4D  43             43  0041A        .ASCII  \CMPH  \
                                71FD  00420        .WORD   29181
                                    02  00422        .BYTE   2
                                    08  00423        .BYTE   8
20  20  4C  50  4D  43             43  00424        .ASCII  \CMPL  \
                                D100  0042A        .WORD   -12032
                                    02  0042C        .BYTE   2
                                    02  0042D        .BYTE   2
```

```
20 33 50 50 4D 43 0042E         .ASCII  \CMPP3 \
            3500 00434          .WORD   13568
              0A 00436          .BYTE   10
              C0 00437          .BYTE   -64
20 34 50 50 4D 43 00438         .ASCII  \CMPP4 \
            3700 0043E          .WORD   14080
              02 00440          .BYTE   2
              0C 00441          .BYTE   12
20 20 56 50 4D 43 00442         .ASCII  \CMPV  \
            EC00 00448          .WORD   -5120
              08 0044A          .BYTE   8
              D2 0044B          .BYTE   -46
20 20 57 50 4D 43 0044C         .ASCII  \CMPW  \
            B100 00452          .WORD   -20224
              02 00454          .BYTE   2
              01 00455          .BYTE   1
20 56 5A 50 4D 43 00456         .ASCII  \CMPZV \
            ED00 0045C          .WORD   -4864
              08 0045E          .BYTE   8
              D2 0045F          .BYTE   -46
20 20 20 43 52 43 00460         .ASCII  \CRC   \
            0B00 00466          .WORD   2816
              16 00468          .BYTE   22
              00 00469          .BYTE   0
20 44 42 54 56 43 0046A         .ASCII  \CVTBD \
            6C00 00470          .WORD   27648
              08 00472          .BYTE   8
              06 00473          .BYTE   6
20 46 42 54 56 43 00474         .ASCII  \CVTBF \
            4C00 0047A          .WORD   19456
              08 0047C          .BYTE   8
              05 0047D          .BYTE   5
20 47 42 54 56 43 0047E         .ASCII  \CVTBG \
            4CFD 00484          .WORD   19709
              08 00486          .BYTE   8
              07 00487          .BYTE   7
20 48 42 54 56 43 00488         .ASCII  \CVTBH \
            6CFD 0048E          .WORD   27901
              08 00490          .BYTE   8
              08 00491          .BYTE   8
20 4C 42 54 56 43 00492         .ASCII  \CVTBL \
            9800 00498          .WORD   -26624
              08 0049A          .BYTE   8
              02 0049B          .BYTE   2
20 57 42 54 56 43 0049C         .ASCII  \CVTBW \
            9900 004A2          .WORD   -26368
              08 004A4          .BYTE   8
              01 004A5          .BYTE   1
20 42 44 54 56 43 004A6         .ASCII  \CVTDB \
            6800 004AC          .WORD   26624
              08 004AE          .BYTE   8
              60 004AF          .BYTE   96
20 46 44 54 56 43 004B0         .ASCII  \CVTDF \
            7600 004B6          .WORD   30208
              08 004B8          .BYTE   8
              65 004B9          .BYTE   101
20 48 44 54 56 43 004BA         .ASCII  \CVTDH \
```

```
                                32FD  004C0          .WORD   13053
                                  08  004C2          .BYTE   8
                                  68  004C3          .BYTE   104
                20  4C  44  54  56  43  004C4          .ASCII  \CVTDL \
                                6A00  004CA          .WORD   27136
                                  08  004CC          .BYTE   8
                                  62  004CD          .BYTE   98
                20  57  44  54  56  43  004CE          .ASCII  \CVTDW \
                                6900  004D4          .WORD   26880
                                  08  004D6          .BYTE   8
                                  61  004D7          .BYTE   97
                20  42  46  54  56  43  004D8          .ASCII  \CVTFB \
                                4800  004DE          .WORD   18432
                                  08  004E0          .BYTE   8
                                  50  004E1          .BYTE   80
                20  44  46  54  56  43  004E2          .ASCII  \CVTFD \
                                5600  004E8          .WORD   22016
                                  08  004EA          .BYTE   8
                                  56  004EB          .BYTE   86
                20  47  46  54  56  43  004EC          .ASCII  \CVTFG \
                                99FD  004F2          .WORD   -26115
                                  08  004F4          .BYTE   8
                                  57  004F5          .BYTE   87
                20  48  46  54  56  43  004F6          .ASCII  \CVTFH \
                                98FD  004FC          .WORD   -26371
                                  08  004FE          .BYTE   8
                                  58  004FF          .BYTE   88
                20  4C  46  54  56  43  00500          .ASCII  \CVTFL \
                                4A00  00506          .WORD   18944
                                  08  00508          .BYTE   8
                                  52  00509          .BYTE   82
                20  57  46  54  56  43  0050A          .ASCII  \CVTFW \
                                4900  00510          .WORD   18688
                                  08  00512          .BYTE   8
                                  51  00513          .BYTE   81
                20  42  47  54  56  43  00514          .ASCII  \CVTGB \
                                48FD  0051A          .WORD   18685
                                  08  0051C          .BYTE   8
                                  70  0051D          .BYTE   112
                20  46  47  54  56  43  0051E          .ASCII  \CVTGF \
                                33FD  00524          .WORD   13309
                                  08  00526          .BYTE   8
                                  75  00527          .BYTE   117
                20  48  47  54  56  43  00528          .ASCII  \CVTGH \
                                56FD  0052E          .WORD   22269
                                  08  00530          .BYTE   8
                                  78  00531          .BYTE   120
                20  4C  47  54  56  43  00532          .ASCII  \CVTGL \
                                4AFD  00538          .WORD   19197
                                  08  0053A          .BYTE   8
                                  72  0053B          .BYTE   114
                20  57  47  54  56  43  0053C          .ASCII  \CVTGW \
                                49FD  00542          .WORD   18941
                                  08  00544          .BYTE   8
                                  71  00545          .BYTE   113
                20  42  48  54  56  43  00546          .ASCII  \CVTHB \
                                68FD  0054C          .WORD   26877
```

```
                                    08  0054E          .BYTE   8
                                    80  0054F          .BYTE   -128
                20  44  48  54  56  43  00550          .ASCII  \CVTHD \
                                  F7FD  00556          .WORD   -2051
                                    08  00558          .BYTE   8
                                    86  00559          .BYTE   -122
                20  46  48  54  56  43  0055A          .ASCII  \CVTHF \
                                  F6FD  00560          .WORD   -2307
                                    08  00562          .BYTE   8
                                    85  00563          .BYTE   -123
                20  47  48  54  56  43  00564          .ASCII  \CVTHG \
                                  76FD  0056A          .WORD   30461
                                    08  0056C          .BYTE   8
                                    87  0056D          .BYTE   -121
                20  4C  48  54  56  43  0056E          .ASCII  \CVTHL \
                                  6AFD  00574          .WORD   27389
                                    08  00576          .BYTE   8
                                    82  00577          .BYTE   -126
                20  57  48  54  56  43  00578          .ASCII  \CVTHW \
                                  69FD  0057E          .WORD   27133
                                    08  00580          .BYTE   8
                                    81  00581          .BYTE   -127
                20  42  4C  54  56  43  00582          .ASCII  \CVTLB \
                                  F600  00588          .WORD   -2560
                                    08  0058A          .BYTE   8
                                    20  0058B          .BYTE   32
                20  44  4C  54  56  43  0058C          .ASCII  \CVTLD \
                                  6E00  00592          .WORD   28160
                                    08  00594          .BYTE   8
                                    26  00595          .BYTE   38
                20  46  4C  54  56  43  00596          .ASCII  \CVTLF \
                                  4E00  0059C          .WORD   19968
                                    08  0059E          .BYTE   8
                                    25  0059F          .BYTE   37
                20  47  4C  54  56  43  005A0          .ASCII  \CVTLG \
                                  4EFD  005A6          .WORD   20221
                                    08  005A8          .BYTE   8
                                    27  005A9          .BYTE   39
                20  48  4C  54  56  43  005AA          .ASCII  \CVTLH \
                                  6EFD  005B0          .WORD   28413
                                    08  005B2          .BYTE   8
                                    28  005B3          .BYTE   40
                20  50  4C  54  56  43  005B4          .ASCII  \CVTLP \
                                  F900  005BA          .WORD   -1792
                                    08  005BC          .BYTE   8
                                    2C  005BD          .BYTE   44
                20  57  4C  54  56  43  005BE          .ASCII  \CVTLW \
                                  F700  005C4          .WORD   -2304
                                    08  005C6          .BYTE   8
                                    21  005C7          .BYTE   33
                20  4C  50  54  56  43  005C8          .ASCII  \CVTPL \
                                  3600  005CE          .WORD   13824
                                    08  005D0          .BYTE   8
                                    C2  005D1          .BYTE   -62
                20  53  50  54  56  43  005D2          .ASCII  \CVTPS \
                                  0800  005D8          .WORD   2048
                                    08  005DA          .BYTE   8
```

```
                         CB   005DB        .BYTE    -53
    20  54  50  54  56   43   005DC        .ASCII   \CVTPT \
                       2400   005E2        .WORD    9216
                         0C   005E4        .BYTE    12
                         CB   005E5        .BYTE    -53
    4C  44  52  54  56   43   005E6        .ASCII   \CVTRDL\
                       6B00   005EC        .WORD    27392
                         08   005EE        .BYTE    8
                         62   005EF        .BYTE    98
    4C  46  52  54  56   43   005F0        .ASCII   \CVTRFL\
                       4B00   005F6        .WORD    19200
                         08   005F8        .BYTE    8
                         52   005F9        .BYTE    82
    4C  47  52  54  56   43   005FA        .ASCII   \CVTRGL\
                       4BFD   00600        .WORD    19453
                         08   00602        .BYTE    8
                         72   00603        .BYTE    114
    4C  48  52  54  56   43   00604        .ASCII   \CVTRHL\
                       6BFD   0060A        .WORD    27645
                         08   0060C        .BYTE    8
                         82   0060D        .BYTE    -126
    20  50  53  54  56   43   0060E        .ASCII   \CVTSP \
                       0900   00614        .WORD    2304
                         08   00616        .BYTE    8
                         BC   00617        .BYTE    -68
    20  50  54  54  56   43   00618        .ASCII   \CVTTP \
                       2600   0061E        .WORD    9728
                         0C   00620        .BYTE    12
                         BC   00621        .BYTE    -68
    20  42  57  54  56   43   00622        .ASCII   \CVTWB \
                       3300   00628        .WORD    13056
                         08   0062A        .BYTE    8
                         10   0062B        .BYTE    16
    20  44  57  54  56   43   0062C        .ASCII   \CVTWD \
                       6D00   00632        .WORD    27904
                         08   00634        .BYTE    8
                         16   00635        .BYTE    22
    20  46  57  54  56   43   00636        .ASCII   \CVTWF \
                       4D00   0063C        .WORD    19712
                         08   0063E        .BYTE    8
                         15   0063F        .BYTE    21
    20  47  57  54  56   43   00640        .ASCII   \CVTWG \
                       4DFD   00646        .WORD    19965
                         08   00648        .BYTE    8
                         17   00649        .BYTE    23
    20  48  57  54  56   43   0064A        .ASCII   \CVTWH \
                       6DFD   00650        .WORD    28157
                         08   00652        .BYTE    8
                         18   00653        .BYTE    24
    20  4C  57  54  56   43   00654        .ASCII   \CVTWL \
                       3200   0065A        .WORD    12800
                         08   0065C        .BYTE    8
                         12   0065D        .BYTE    18
    20  20  42  43  45   44   0065E        .ASCII   \DECB  \
                       9700   00664        .WORD    -26880
                         01   00666        .BYTE    1
                         00   00667        .BYTE    0
```

```
20  20  4C  43  45  44  00668          .ASCII  \DECL  \
                D700    0066E          .WORD   -10496
                  01    00670          .BYTE   1
                  02    00671          .BYTE   2
20  20  57  43  45  44  00672          .ASCII  \DECW  \
                B700    00678          .WORD   -18688
                  01    0067A          .BYTE   1
                  01    0067B          .BYTE   1
20  32  42  56  49  44  0067C          .ASCII  \DIVB2 \
                8600    00682          .WORD   -31232
                  02    00684          .BYTE   2
                  00    00685          .BYTE   0
20  33  42  56  49  44  00686          .ASCII  \DIVB3 \
                8700    0068C          .WORD   -30976
                  03    0068E          .BYTE   3
                  00    0068F          .BYTE   0
20  32  44  56  49  44  00690          .ASCII  \DIVD2 \
                6600    00696          .WORD   26112
                  02    00698          .BYTE   2
                  06    00699          .BYTE   6
20  33  44  56  49  44  0069A          .ASCII  \DIVD3 \
                6700    006A0          .WORD   26368
                  03    006A2          .BYTE   3
                  06    006A3          .BYTE   6
20  32  46  56  49  44  006A4          .ASCII  \DIVF2 \
                4600    006AA          .WORD   17920
                  02    006AC          .BYTE   2
                  05    006AD          .BYTE   5
20  33  46  56  49  44  006AE          .ASCII  \DIVF3 \
                4700    006B4          .WORD   18176
                  03    006B6          .BYTE   3
                  05    006B7          .BYTE   5
20  32  47  56  49  44  006B8          .ASCII  \DIVG2 \
                46FD    006BE          .WORD   18173
                  02    006C0          .BYTE   2
                  07    006C1          .BYTE   7
20  33  47  56  49  44  006C2          .ASCII  \DIVG3 \
                47FD    006C8          .WORD   18429
                  03    006CA          .BYTE   3
                  07    006CB          .BYTE   7
20  32  48  56  49  44  006CC          .ASCII  \DIVH2 \
                66FD    006D2          .WORD   26365
                  02    006D4          .BYTE   2
                  08    006D5          .BYTE   8
20  33  48  56  49  44  006D6          .ASCII  \DIVH3 \
                67FD    006DC          .WORD   26621
                  03    006DE          .BYTE   3
                  08    006DF          .BYTE   8
20  32  4C  56  49  44  006E0          .ASCII  \DIVL2 \
                C600    006E6          .WORD   -14848
                  02    006E8          .BYTE   2
                  02    006E9          .BYTE   2
20  33  4C  56  49  44  006EA          .ASCII  \DIVL3 \
                C700    006F0          .WORD   -14592
                  03    006F2          .BYTE   3
                  02    006F3          .BYTE   2
20  20  50  56  49  44  006F4          .ASCII  \DIVP  \
```

```
                                2700  006FA          .WORD   9984
                                  03  006FC          .BYTE   3
                                  0C  006FD          .BYTE   12
                20  32  57  56  49  44  006FE          .ASCII  \DIVW2 \
                              A600  00704          .WORD   -23040
                                  02  00706          .BYTE   2
                                  01  00707          .BYTE   1
                20  33  57  56  49  44  00708          .ASCII  \DIVW3 \
                              A700  0070E          .WORD   -22784
                                  03  00710          .BYTE   3
                                  01  00711          .BYTE   1
                43  50  54  49  44  45  00712          .ASCII  \EDITPC\
                              3800  00718          .WORD   14336
                                  0B  0071A          .BYTE   11
                                  C0  0071B          .BYTE   -64
                20  20  56  49  44  45  0071C          .ASCII  \EDIV  \
                              7B00  00722          .WORD   31488
                                  12  00724          .BYTE   18
                                  00  00725          .BYTE   0
                20  44  44  4F  4D  45  00726          .ASCII  \EMODD \
                              7400  0072C          .WORD   29696
                                  13  0072E          .BYTE   19
                                  96  0072F          .BYTE   -106
                20  46  44  4F  4D  45  00730          .ASCII  \EMODF \
                              5400  00736          .WORD   21504
                                  13  00738          .BYTE   19
                                  95  00739          .BYTE   -107
                20  47  44  4F  4D  45  0073A          .ASCII  \EMODG \
                              54FD  00740          .WORD   21757
                                  13  00742          .BYTE   19
                                  A7  00743          .BYTE   -89
                20  48  44  4F  4D  45  00744          .ASCII  \EMODH \
                              74FD  0074A          .WORD   29949
                                  13  0074C          .BYTE   19
                                  A8  0074D          .BYTE   -88
                20  20  4C  55  4D  45  0074E          .ASCII  \EMUL  \
                              7A00  00754          .WORD   31232
                                  14  00756          .BYTE   20
                                  00  00757          .BYTE   0
                20  20  56  54  58  45  00758          .ASCII  \EXTV  \
                              EE00  0075E          .WORD   -4608
                                  08  00760          .BYTE   8
                                  D2  00761          .BYTE   -46
                20  56  5A  54  58  45  00762          .ASCII  \EXTZV \
                              EF00  00768          .WORD   -4352
                                  08  0076A          .BYTE   8
                                  D2  0076B          .BYTE   -46
                20  20  20  43  46  46  0076C          .ASCII  \FFC   \
                              EB00  00772          .WORD   -5376
                                  08  00774          .BYTE   8
                                  D2  00775          .BYTE   -46
                20  20  20  53  46  46  00776          .ASCII  \FFS   \
                              EA00  0077C          .WORD   -5632
                                  08  0077E          .BYTE   8
                                  D2  0077F          .BYTE   -46
                20  20  54  4C  41  4C  00780          .ASCII  \HALT  \
                              0000  00786          .WORD   0
```

```
                                00  00788          .BYTE   0
                                00  00789          .BYTE   0
20  20  42  43  4E  49           0078A          .ASCII  \INCB  \
                          9600  00790          .WORD   -27136
                            01  00792          .BYTE   1
                            00  00793          .BYTE   0
20  20  4C  43  4E  49           00794          .ASCII  \INCL  \
                          D600  0079A          .WORD   -10752
                            01  0079C          .BYTE   1
                            02  0079D          .BYTE   2
20  20  57  43  4E  49           0079E          .ASCII  \INCW  \
                          B600  007A4          .WORD   -18944
                            01  007A6          .BYTE   1
                            01  007A7          .BYTE   1
20  58  45  44  4E  49           007A8          .ASCII  \INDEX \
                          0A00  007AE          .WORD   2560
                            15  007B0          .BYTE   21
                            00  007B1          .BYTE   0
49  48  51  53  4E  49           007B2          .ASCII  \INSQHI\
                          5C00  007B8          .WORD   23552
                            02  007BA          .BYTE   2
                            00  007BB          .BYTE   0
49  54  51  53  4E  49           007BC          .ASCII  \INSQTI\
                          5D00  007C2          .WORD   23808
                            02  007C4          .BYTE   2
                            00  007C5          .BYTE   0
45  55  51  53  4E  49           007C6          .ASCII  \INSQUE\
                          0E00  007CC          .WORD   3584
                            02  007CE          .BYTE   2
                            00  007CF          .BYTE   0
20  20  56  53  4E  49           007D0          .ASCII  \INSV  \
                          F000  007D6          .WORD   -4096
                            08  007D8          .BYTE   8
                            2D  007D9          .BYTE   45
20  20  20  50  4D  4A           007DA          .ASCII  \JMP   \
                          1700  007E0          .WORD   5888
                            01  007E2          .BYTE   1
                            00  007E3          .BYTE   0
20  20  20  42  53  4A           007E4          .ASCII  \JSB   \
                          1600  007EA          .WORD   5632
                            01  007EC          .BYTE   1
                            00  007ED          .BYTE   0
58  54  43  50  44  4C           007EE          .ASCII  \LDPCTX\
                          0600  007F4          .WORD   1536
                            00  007F6          .BYTE   0
                            00  007F7          .BYTE   0
20  20  43  43  4F  4C           007F8          .ASCII  \LOCC  \
                          3A00  007FE          .WORD   14848
                            08  00800          .BYTE   8
                            0B  00801          .BYTE   11
43  48  43  54  41  4D           00802          .ASCII  \MATCHC\
                          3900  00808          .WORD   14592
                            02  0080A          .BYTE   2
                            0B  0080B          .BYTE   11
20  42  4D  4F  43  4D           0080C          .ASCII  \MCOMB \
                          9200  00812          .WORD   -28160
                            02  00814          .BYTE   2
```

```
                                    00  00815          .BYTE   0
20  4C  4D  4F  43  4D              4D  00816          .ASCII  \MCOML \
                                  D200  0081C          .WORD   -11776
                                    02  0081E          .BYTE   2
                                    02  0081F          .BYTE   2
20  57  4D  4F  43  4D              4D  00820          .ASCII  \MCOMW \
                                  B200  00826          .WORD   -19968
                                    02  00828          .BYTE   2
                                    01  00829          .BYTE   1
20  20  52  50  46  4D              4D  0082A          .ASCII  \MFPR  \
                                  DB00  00830          .WORD   -9472
                                    02  00832          .BYTE   2
                                    02  00833          .BYTE   2
20  42  47  45  4E  4D              4D  00834          .ASCII  \MNEGB \
                                  8E00  0083A          .WORD   -29184
                                    02  0083C          .BYTE   2
                                    00  0083D          .BYTE   0
20  44  47  45  4E  4D              4D  0083E          .ASCII  \MNEGD \
                                  7200  00844          .WORD   29184
                                    02  00846          .BYTE   2
                                    06  00847          .BYTE   6
20  46  47  45  4E  4D              4D  00848          .ASCII  \MNEGF \
                                  5200  0084E          .WORD   20992
                                    02  00850          .BYTE   2
                                    05  00851          .BYTE   5
20  47  47  45  4E  4D              4D  00852          .ASCII  \MNEGG \
                                  52FD  00858          .WORD   21245
                                    02  0085A          .BYTE   2
                                    07  0085B          .BYTE   7
20  48  47  45  4E  4D              4D  0085C          .ASCII  \MNEGH \
                                  72FD  00862          .WORD   29437
                                    02  00864          .BYTE   2
                                    08  00865          .BYTE   8
20  4C  47  45  4E  4D              4D  00866          .ASCII  \MNEGL \
                                  CE00  0086C          .WORD   -12800
                                    02  0086E          .BYTE   2
                                    02  0086F          .BYTE   2
20  57  47  45  4E  4D              4D  00870          .ASCII  \MNEGW \
                                  AE00  00876          .WORD   -20992
                                    02  00878          .BYTE   2
                                    01  00879          .BYTE   1
20  42  41  56  4F  4D              4D  0087A          .ASCII  \MOVAB \
                                  9E00  00880          .WORD   -25088
                                    08  00882          .BYTE   8
                                    02  00883          .BYTE   2
20  44  41  56  4F  4D              4D  00884          .ASCII  \MOVAD \
                                  7E00  0088A          .WORD   32256
                                    08  0088C          .BYTE   8
                                    62  0088D          .BYTE   98
20  46  41  56  4F  4D              4D  0088E          .ASCII  \MOVAF \
                                  DE00  00894          .WORD   -8704
                                    08  00896          .BYTE   8
                                    52  00897          .BYTE   82
20  47  41  56  4F  4D              4D  00898          .ASCII  \MOVAG \
                                  7E00  0089E          .WORD   32256
                                    08  008A0          .BYTE   8
                                    72  008A1          .BYTE   114
```

```
                20  48  41  56  4F  4D  008A2          .ASCII  \MOVAH \
                                7EFD    008A8          .WORD   32509
                                  08    008AA          .BYTE   8
                                  82    008AB          .BYTE   -126
                20  4C  41  56  4F  4D  008AC          .ASCII  \MOVAL \
                                DE00    008B2          .WORD   -8704
                                  08    008B4          .BYTE   8
                                  22    008B5          .BYTE   34
                20  4F  41  56  4F  4D  008B6          .ASCII  \MOVAO \
                                7EFD    008BC          .WORD   32509
                                  08    008BE          .BYTE   8
                                  42    008BF          .BYTE   66
                20  51  41  56  4F  4D  008C0          .ASCII  \MOVAQ \
                                7E00    008C6          .WORD   32256
                                  08    008C8          .BYTE   8
                                  32    008C9          .BYTE   50
                20  57  41  56  4F  4D  008CA          .ASCII  \MOVAW \
                                3E00    008D0          .WORD   15872
                                  08    008D2          .BYTE   8
                                  12    008D3          .BYTE   18
                20  20  42  56  4F  4D  008D4          .ASCII  \MOVB  \
                                9000    008DA          .WORD   -28672
                                  02    008DC          .BYTE   2
                                  00    008DD          .BYTE   0
                20  33  43  56  4F  4D  008DE          .ASCII  \MOVC3 \
                                2800    008E4          .WORD   10240
                                  0A    008E6          .BYTE   10
                                  B0    008E7          .BYTE   -80
                20  35  43  56  4F  4D  008E8          .ASCII  \MOVC5 \
                                2C00    008EE          .WORD   11264
                                  0C    008F0          .BYTE   12
                                  BB    008F1          .BYTE   -69
                20  20  44  56  4F  4D  008F2          .ASCII  \MOVD  \
                                7000    008F8          .WORD   28672
                                  02    008FA          .BYTE   2
                                  06    008FB          .BYTE   6
                20  20  46  56  4F  4D  008FC          .ASCII  \MOVF  \
                                5000    00902          .WORD   20480
                                  02    00904          .BYTE   2
                                  05    00905          .BYTE   5
                20  20  47  56  4F  4D  00906          .ASCII  \MOVG  \
                                50FD    0090C          .WORD   20733
                                  02    0090E          .BYTE   2
                                  07    0090F          .BYTE   7
                20  20  48  56  4F  4D  00910          .ASCII  \MOVH  \
                                70FD    00916          .WORD   28925
                                  02    00918          .BYTE   2
                                  08    00919          .BYTE   8
                20  20  4C  56  4F  4D  0091A          .ASCII  \MOVL  \
                                D000    00920          .WORD   -12288
                                  02    00922          .BYTE   2
                                  02    00923          .BYTE   2
                20  20  4F  56  4F  4D  00924          .ASCII  \MOVO  \
                                7DFD    0092A          .WORD   32253
                                  02    0092C          .BYTE   2
                                  04    0092D          .BYTE   4
                20  20  50  56  4F  4D  0092E          .ASCII  \MOVP  \
```

```
                        3400  00934          .WORD   13312
                          0A  00936          .BYTE   10
                          C0  00937          .BYTE   -64
4C  53  50  56  4F  4D        00938          .ASCII  \MOVPSL\
                        DC00  0093E          .WORD   -9216
                          01  00940          .BYTE   1
                          02  00941          .BYTE   2
20  20  51  56  4F  4D        00942          .ASCII  \MOVQ  \
                        7D00  00948          .WORD   32000
                          02  0094A          .BYTE   2
                          03  0094B          .BYTE   3
20  43  54  56  4F  4D        0094C          .ASCII  \MOVTC \
                        2E00  00952          .WORD   11776
                          0D  00954          .BYTE   13
                          BB  00955          .BYTE   -69
43  55  54  56  4F  4D        00956          .ASCII  \MOVTUC\
                        2F00  0095C          .WORD   12032
                          0D  0095E          .BYTE   13
                          BB  0095F          .BYTE   -69
20  20  57  56  4F  4D        00960          .ASCII  \MOVW  \
                        B000  00966          .WORD   -20480
                          02  00968          .BYTE   2
                          01  00969          .BYTE   1
4C  42  5A  56  4F  4D        0096A          .ASCII  \MOVZBL\
                        9A00  00970          .WORD   -26112
                          08  00972          .BYTE   8
                          02  00973          .BYTE   2
57  42  5A  56  4F  4D        00974          .ASCII  \MOVZBW\
                        9B00  0097A          .WORD   -25856
                          08  0097C          .BYTE   8
                          01  0097D          .BYTE   1
4C  57  5A  56  4F  4D        0097E          .ASCII  \MOVZWL\
                        3C00  00984          .WORD   15360
                          08  00986          .BYTE   8
                          12  00987          .BYTE   18
20  20  52  50  54  4D        00988          .ASCII  \MTPR  \
                        DA00  0098E          .WORD   -9728
                          02  00990          .BYTE   2
                          02  00991          .BYTE   2
20  32  42  4C  55  4D        00992          .ASCII  \MULB2 \
                        8400  00998          .WORD   -31744
                          02  0099A          .BYTE   2
                          00  0099B          .BYTE   0
20  33  42  4C  55  4D        0099C          .ASCII  \MULB3 \
                        8500  009A2          .WORD   -31488
                          03  009A4          .BYTE   3
                          00  009A5          .BYTE   0
20  32  44  4C  55  4D        009A6          .ASCII  \MULD2 \
                        6400  009AC          .WORD   25600
                          02  009AE          .BYTE   2
                          06  009AF          .BYTE   6
20  33  44  4C  55  4D        009B0          .ASCII  \MULD3 \
                        6500  009B6          .WORD   25856
                          03  009B8          .BYTE   3
                          06  009B9          .BYTE   6
20  32  46  4C  55  4D        009BA          .ASCII  \MULF2 \
                        4400  009C0          .WORD   17408
```

```
                                    02  009C2          .BYTE   2
                                    05  009C3          .BYTE   5
                20  33  46  4C  55  4D  009C4          .ASCII  \MULF3 \
                                  4500  009CA          .WORD   17664
                                    03  009CC          .BYTE   3
                                    05  009CD          .BYTE   5
                20  32  47  4C  55  4D  009CE          .ASCII  \MULG2 \
                                  44FD  009D4          .WORD   17661
                                    02  009D6          .BYTE   2
                                    07  009D7          .BYTE   7
                20  33  47  4C  55  4D  009D8          .ASCII  \MULG3 \
                                  45FD  009DE          .WORD   17917
                                    03  009E0          .BYTE   3
                                    07  009E1          .BYTE   7
                20  32  48  4C  55  4D  009E2          .ASCII  \MULH2 \
                                  64FD  009E8          .WORD   25853
                                    02  009EA          .BYTE   2
                                    08  009EB          .BYTE   8
                20  33  48  4C  55  4D  009EC          .ASCII  \MULH3 \
                                  65FD  009F2          .WORD   26109
                                    03  009F4          .BYTE   3
                                    08  009F5          .BYTE   8
                20  32  4C  4C  55  4D  009F6          .ASCII  \MULL2 \
                                  C400  009FC          .WORD   -15360
                                    02  009FE          .BYTE   2
                                    02  009FF          .BYTE   2
                20  33  4C  4C  55  4D  00A00          .ASCII  \MULL3 \
                                  C500  00A06          .WORD   -15104
                                    03  00A08          .BYTE   3
                                    02  00A09          .BYTE   2
                20  20  50  4C  55  4D  00A0A          .ASCII  \MULP  \
                                  2500  00A10          .WORD   9472
                                    03  00A12          .BYTE   3
                                    0C  00A13          .BYTE   12
                20  32  57  4C  55  4D  00A14          .ASCII  \MULW2 \
                                  A400  00A1A          .WORD   -23552
                                    02  00A1C          .BYTE   2
                                    01  00A1D          .BYTE   1
                20  33  57  4C  55  4D  00A1E          .ASCII  \MULW3 \
                                  A500  00A24          .WORD   -23296
                                    03  00A26          .BYTE   3
                                    01  00A27          .BYTE   1
                20  20  20  50  4F  4E  00A28          .ASCII  \NOP   \
                                  0100  00A2E          .WORD   256
                                    00  00A30          .BYTE   0
                                    00  00A31          .BYTE   0
                20  44  59  4C  4F  50  00A32          .ASCII  \POLYD \
                                  7500  00A38          .WORD   29952
                                    08  00A3A          .BYTE   8
                                    6B  00A3B          .BYTE   107
                20  46  59  4C  4F  50  00A3C          .ASCII  \POLYF \
                                  5500  00A42          .WORD   21760
                                    08  00A44          .BYTE   8
                                    5B  00A45          .BYTE   91
                20  47  59  4C  4F  50  00A46          .ASCII  \POLYG \
                                  55FD  00A4C          .WORD   22013
                                    08  00A4E          .BYTE   8
```

```
                        7B  00A4F       .BYTE   123
  20  48  59  4C  4F    50  00A50       .ASCII  \POLYH \
                      75FD  00A56       .WORD   30205
                        08  00A58       .BYTE   8
                        8B  00A59       .BYTE   -117
  20  20  52  50  4F    50  00A5A       .ASCII  \POPR  \
                      BA00  00A60       .WORD   -17920
                        01  00A62       .BYTE   1
                        01  00A63       .BYTE   1
  52  45  42  4F  52    50  00A64       .ASCII  \PROBER\
                      0C00  00A6A       .WORD   3072
                        08  00A6C       .BYTE   8
                        0B  00A6D       .BYTE   11
  57  45  42  4F  52    50  00A6E       .ASCII  \PROBEW\
                      0D00  00A74       .WORD   3328
                        08  00A76       .BYTE   8
                        0B  00A77       .BYTE   11
  42  41  48  53  55    50  00A78       .ASCII  \PUSHAB\
                      9F00  00A7E       .WORD   -24832
                        01  00A80       .BYTE   1
                        00  00A81       .BYTE   0
  44  41  48  53  55    50  00A82       .ASCII  \PUSHAD\
                      7F00  00A88       .WORD   32512
                        01  00A8A       .BYTE   1
                        06  00A8B       .BYTE   6
  46  41  48  53  55    50  00A8C       .ASCII  \PUSHAF\
                      DF00  00A92       .WORD   -8448
                        01  00A94       .BYTE   1
                        05  00A95       .BYTE   5
  47  41  48  53  55    50  00A96       .ASCII  \PUSHAG\
                      7F00  00A9C       .WORD   32512
                        01  00A9E       .BYTE   1
                        07  00A9F       .BYTE   7
  48  41  48  53  55    50  00AA0       .ASCII  \PUSHAH\
                      7FFD  00AA6       .WORD   32765
                        01  00AA8       .BYTE   1
                        08  00AA9       .BYTE   8
  4C  41  48  53  55    50  00AAA       .ASCII  \PUSHAL\
                      DF00  00AB0       .WORD   -8448
                        01  00AB2       .BYTE   1
                        02  00AB3       .BYTE   2
  4F  41  48  53  55    50  00AB4       .ASCII  \PUSHAO\
                      7FFD  00ABA       .WORD   32765
                        01  00ABC       .BYTE   1
                        04  00ABD       .BYTE   4
  51  41  48  53  55    50  00ABE       .ASCII  \PUSHAQ\
                      7F00  00AC4       .WORD   32512
                        01  00AC6       .BYTE   1
                        03  00AC7       .BYTE   3
  57  41  48  53  55    50  00AC8       .ASCII  \PUSHAW\
                      3F00  00ACE       .WORD   16128
                        01  00AD0       .BYTE   1
                        01  00AD1       .BYTE   1
  20  4C  48  53  55    50  00AD2       .ASCII  \PUSHL \
                      DD00  00AD8       .WORD   -8960
                        01  00ADA       .BYTE   1
                        02  00ADB       .BYTE   2
```

```
                20  52  48  53  55  50  00ADC          .ASCII  \PUSHR \
                                BB00    00AE2          .WORD   -17664
                                  01    00AE4          .BYTE   1
                                  01    00AE5          .BYTE   1
                20  20  20  49  45  52  00AE6          .ASCII  \REI   \
                                0200    00AEC          .WORD   512
                                  00    00AEE          .BYTE   0
                                  00    00AEF          .BYTE   0
                49  48  51  4D  45  52  00AF0          .ASCII  \REMQHI\
                                5E00    00AF6          .WORD   24064
                                  02    00AF8          .BYTE   2
                                  00    00AF9          .BYTE   0
                49  54  51  4D  45  52  00AFA          .ASCII  \REMQTI\
                                5F00    00B00          .WORD   24320
                                  02    00B02          .BYTE   2
                                  00    00B03          .BYTE   0
                45  55  51  4D  45  52  00B04          .ASCII  \REMQUE\
                                0F00    00B0A          .WORD   3840
                                  02    00B0C          .BYTE   2
                                  00    00B0D          .BYTE   0
                20  20  20  54  45  52  00B0E          .ASCII  \RET   \
                                0400    00B14          .WORD   1024
                                  00    00B16          .BYTE   0
                                  00    00B17          .BYTE   0
                20  20  4C  54  4F  52  00B18          .ASCII  \ROTL  \
                                9C00    00B1E          .WORD   -25600
                                  10    00B20          .BYTE   16
                                  02    00B21          .BYTE   2
                20  20  20  42  53  52  00B22          .ASCII  \RSB   \
                                0500    00B28          .WORD   1280
                                  00    00B2A          .BYTE   0
                                  00    00B2B          .BYTE   0
                20  20  43  57  42  53  00B2C          .ASCII  \SBWC  \
                                D900    00B32          .WORD   -9984
                                  02    00B34          .BYTE   2
                                  02    00B35          .BYTE   2
                20  43  4E  41  43  53  00B36          .ASCII  \SCANC \
                                2A00    00B3C          .WORD   10752
                                  0B    00B3E          .BYTE   11
                                  B0    00B3F          .BYTE   -80
                20  20  43  50  4B  53  00B40          .ASCII  \SKPC  \
                                3B00    00B46          .WORD   15104
                                  08    00B48          .BYTE   8
                                  0B    00B49          .BYTE   11
                51  45  47  42  4F  53  00B4A          .ASCII  \SOBGEQ\
                                F400    00B50          .WORD   -3072
                                  05    00B52          .BYTE   5
                                  20    00B53          .BYTE   32
                52  54  47  42  4F  53  00B54          .ASCII  \SOBGTR\
                                F500    00B5A          .WORD   -2816
                                  05    00B5C          .BYTE   5
                                  20    00B5D          .BYTE   32
                20  43  4E  41  50  53  00B5E          .ASCII  \SPANC \
                                2B00    00B64          .WORD   11008
                                  0B    00B66          .BYTE   11
                                  B0    00B67          .BYTE   -80
                20  32  42  42  55  53  00B68          .ASCII  \SUBB2 \
```

```
                                8200  00B6E          .WORD   -32256
                                  02  00B70          .BYTE   2
                                  00  00B71          .BYTE   0
20  33  42  42  55  53            53  00B72          .ASCII  \SUBB3 \
                                8300  00B78          .WORD   -32000
                                  03  00B7A          .BYTE   3
                                  00  00B7B          .BYTE   0
20  32  44  42  55  53            53  00B7C          .ASCII  \SUBD2 \
                                6200  00B82          .WORD   25088
                                  02  00B84          .BYTE   2
                                  06  00B85          .BYTE   6
20  33  44  42  55  53            53  00B86          .ASCII  \SUBD3 \
                                6300  00B8C          .WORD   25344
                                  03  00B8E          .BYTE   3
                                  06  00B8F          .BYTE   6
20  32  46  42  55  53            53  00B90          .ASCII  \SUBF2 \
                                4200  00B96          .WORD   16896
                                  02  00B98          .BYTE   2
                                  05  00B99          .BYTE   5
20  33  46  42  55  53            53  00B9A          .ASCII  \SUBF3 \
                                4300  00BA0          .WORD   17152
                                  03  00BA2          .BYTE   3
                                  05  00BA3          .BYTE   5
20  32  47  42  55  53            53  00BA4          .ASCII  \SUBG2 \
                                42FD  00BAA          .WORD   17149
                                  02  00BAC          .BYTE   2
                                  07  00BAD          .BYTE   7
20  33  47  42  55  53            53  00BAE          .ASCII  \SUBG3 \
                                43FD  00BB4          .WORD   17405
                                  03  00BB6          .BYTE   3
                                  07  00BB7          .BYTE   7
20  32  48  42  55  53            53  00BB8          .ASCII  \SUBH2 \
                                62FD  00BBE          .WORD   25341
                                  02  00BC0          .BYTE   2
                                  08  00BC1          .BYTE   8
20  33  48  42  55  53            53  00BC2          .ASCII  \SUBH3 \
                                63FD  00BC8          .WORD   25597
                                  03  00BCA          .BYTE   3
                                  08  00BCB          .BYTE   8
20  32  4C  42  55  53            53  00BCC          .ASCII  \SUBL2 \
                                C200  00BD2          .WORD   -15872
                                  02  00BD4          .BYTE   2
                                  02  00BD5          .BYTE   2
20  33  4C  42  55  53            53  00BD6          .ASCII  \SUBL3 \
                                C300  00BDC          .WORD   -15616
                                  03  00BDE          .BYTE   3
                                  02  00BDF          .BYTE   2
20  34  50  42  55  53            53  00BE0          .ASCII  \SUBP4 \
                                2200  00BE6          .WORD   8704
                                  02  00BE8          .BYTE   2
                                  0C  00BE9          .BYTE   12
20  36  50  42  55  53            53  00BEA          .ASCII  \SUBP6 \
                                2300  00BF0          .WORD   8960
                                  03  00BF2          .BYTE   3
                                  0C  00BF3          .BYTE   12
20  32  57  42  55  53            53  00BF4          .ASCII  \SUBW2 \
                                A200  00BFA          .WORD   -24064
```

```
                                    02  00BFC          .BYTE   2
                                    01  00BFD          .BYTE   1
                20  33  57  42  55  53  00BFE          .ASCII  \SUBW3 \
                                  A300  00C04          .WORD   -23808
                                    03  00C06          .BYTE   3
                                    01  00C07          .BYTE   1
                58  54  43  50  56  53  00C08          .ASCII  \SVPCTX\
                                  0700  00C0E          .WORD   1792
                                    00  00C10          .BYTE   0
                                    00  00C11          .BYTE   0
                20  20  42  54  53  54  00C12          .ASCII  \TSTB  \
                                  9500  00C18          .WORD   -27392
                                    01  00C1A          .BYTE   1
                                    00  00C1B          .BYTE   0
                20  20  44  54  53  54  00C1C          .ASCII  \TSTD  \
                                  7300  00C22          .WORD   29440
                                    01  00C24          .BYTE   1
                                    06  00C25          .BYTE   6
                20  20  46  54  53  54  00C26          .ASCII  \TSTF  \
                                  5300  00C2C          .WORD   21248
                                    01  00C2E          .BYTE   1
                                    05  00C2F          .BYTE   5
                20  20  47  54  53  54  00C30          .ASCII  \TSTG  \
                                  53FD  00C36          .WORD   21501
                                    01  00C38          .BYTE   1
                                    07  00C39          .BYTE   7
                20  20  48  54  53  54  00C3A          .ASCII  \TSTH  \
                                  73FD  00C40          .WORD   29693
                                    01  00C42          .BYTE   1
                                    08  00C43          .BYTE   8
                20  20  4C  54  53  54  00C44          .ASCII  \TSTL  \
                                  D500  00C4A          .WORD   -11008
                                    01  00C4C          .BYTE   1
                                    02  00C4D          .BYTE   2
                20  20  57  54  53  54  00C4E          .ASCII  \TSTW  \
                                  B500  00C54          .WORD   -19200
                                    01  00C56          .BYTE   1
                                    01  00C57          .BYTE   1
                20  20  20  43  46  58  00C58          .ASCII  \XFC   \
                                  FC00  00C5E          .WORD   -1024
                                    00  00C60          .BYTE   0
                                    00  00C61          .BYTE   0
                20  32  42  52  4F  58  00C62          .ASCII  \XORB2 \
                                  8C00  00C68          .WORD   -29696
                                    02  00C6A          .BYTE   2
                                    00  00C6B          .BYTE   0
                20  33  42  52  4F  58  00C6C          .ASCII  \XORB3 \
                                  8D00  00C72          .WORD   -29440
                                    03  00C74          .BYTE   3
                                    00  00C75          .BYTE   0
                20  32  4C  52  4F  58  00C76          .ASCII  \XORL2 \
                                  CC00  00C7C          .WORD   -13312
                                    02  00C7E          .BYTE   2
                                    02  00C7F          .BYTE   2
                20  33  4C  52  4F  58  00C80          .ASCII  \XORL3 \
                                  CD00  00C86          .WORD   -13056
                                    03  00C88          .BYTE   3
```

```
                                                                02  00C89           .BYTE   2
                                    20   32   57   52   4F      58  00C8A           .ASCII  \XORW2 \
                                                              AC00  00C90           .WORD   -21504
                                                                02  00C92           .BYTE   2
                                                                01  00C93           .BYTE   1
                                    20   33   57   52   4F      58  00C94           .ASCII  \XORW3 \
                                                              AD00  00C9A           .WORD   -21248
                                                                03  00C9C           .BYTE   3
                                                                01  00C9D           .BYTE   1
                                                                    00C9E           .BLKB   2
009B  0095  0134  00CB  011D  011B  0048  0117  0104  00C0  00CA0 DBG$OPCODE_KIND_TABLE::
                                                                                    .WORD   192, 260, 279, 72, 283, 285, 203, 308, -
0029  0046  0049  004B  011A  00C7  010B  010A  0070  00C4  00CB4                   149, 155, 196, 112, 266, 267, 199, 282, -
0050  004F  0043  002E  0044  002B  00C9  00CA  0042  002D  00CC8                   75, 73, 70, 41, 45, 66, 202, 201, 43, 68, -
00B2  009C  0101  0096  0131  0130  0016  0015  0045  002C  00CDC                   46, 67, 79, 80, 44, 69, 21, 22, 304, 305, -
004A  004C  00EF  00EE  0065  00E4  0123  011F  0064  00E3  00CF0                   150, 257, 156, 178, 227, 100, 287, 291, -
0120  00CC  00CD  00B5  006C  0094  006B  00EB  009D  00A2  00D04                   228, 101, 238, 239, 76, 74, 162, 157, -
00FA  00F9  0129  0128  000E  000D  0114  00E1  0007  00F3  00D18                   235, 107, 148, 108, 181, 205, 204, 288, -
0003  008F  009F  0072  0098  0080  0081  007C  00AB  00AA  00D2C                   243, 7, 225, 276, 13, 14, 296, 297, 249, -
                  007D  0106  00B8  0137  00D4  0067  00E6  00D40                   250, 170, 171, 124, 129, 128, 152, 114, -
                                                                                    159, 143, 3, 230, 103, 212, 311, 184, -
                                                                                    262, 125
                                                               00#  00D4E           .BYTE   0[2]
                                                              0008  00D50           .WORD   8
                                                               00#  00D52           .BYTE   0[6]
00F8  00F7  0127  0126  000C  000B  0119  0118  00C6  00C5  00D58           .WORD   197, 198, 280, 281, 11, 12, 294, 295, -
0002  008E  009E  0071  0097  007A  0C7B  0077  00A9  00A8  00D6C                   247, 248, 168, 169, 119, 123, 122, 151, -
                  0078  0105  00B7  0136  00D3  0066  00E5  00D80                   113, 158, 142, 2, 229, 102, 211, 310, -
                                                                                    183, 261, 120
                                                               00#  00D8E           .BYTE   0[2]
000A  0009  0113  00E0  00ED  0061  00B6  00BB  001E  001C  00D90           .WORD   28, 30, 187, 182, 97, 237, 224, 275, 9, -
0030  002F  0037  0036  00A7  00A6  00F6  00F5  0125  0124  00DA4                   10, 292, 293, 245, 246, 166, 167, 54, 55, -
0135  005A  003D  00CE  0063  00E2  0053  00D2  013E  013D  00DB8                   47, 48, 317, 318, 210, 83, 226, 99, 206, -
010C  00D9  0001  011C  00F2  00F1  0076  0075  00A3  00C1  00DCC                   61, 90, 309, 193, 163, 117, 118, 241, -
003C  003B  00B4  00B3  0103  0102  0133  0132  0018  0017  00DE0                   242, 284, 1, 217, 268, 23, 24, 306, 307, -
003F  00D0  006E  00F0  0055  00D8  0142  0141  0035  0034  00DF4                   258, 259, 179, 180, 59, 60, 52, 53, 321, -
0056  0057  0116  0109  0033  003A  00A5  00C3  013B  0062  00E08                   322, 216, 85, 240, 110, 208, 63, 98, 315, -
00B1  00B0  0100  00FF  012F  012E  0014  0013  0059  0058  00E1C                   195, 165, 58, 51, 265, 278, 87, 86, 88, -
006A  00E9  0054  00D7  0140  013F  0032  0031  0039  0038  00E30                   89, 19, 20, 302, 303, 255, 256, 176, 177, -
00D1  00F4  011E  0019  00A4  00C2  013A  005F  003E  00CF  00E44                   56, 57, 49, 50, 319, 320, 215, 84, 233, -
0020  0024  0022  0025  001F  0023  0111  00DE  0115  00EC  00E58                   106, 207, 62, 95, 314, 194, 164, 25, 286, -
00BD  00BC  006F  006D  00BE  00BF  0040  0041  0021  0026  00E6C                   244, 209, 236, 277, 222, 273, 35, 31, 37, -
0092  001D  0093  008D  0122  0121  001A  001B  0006  00C8  00E80                   34, 36, 32, 38, 33, 65, 64, 191, 190, -
                                          013C  0052  0051  00E94                   109, 111, 188, 189, 200, 6, 27, 26, 289, -
                                                                                    290, 141, 147, 29, 146, 81, 82, 316
                                                               00#  00E9A           .BYTE   0[6]
0000  0000  0000  0000  0000  0000  0000  0000  0000  0000  00EA0           .WORD   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -
                        0000  0000  0000  0000  0000  0000  00EB4                   0, 0
                                                               00#  00EC0           .BYTE   0[68]
                                                        0083  0079  00F04           .WORD   121, 131
                                                               00#  00F08           .BYTE   0[24]
0086  0082  00AD  00AC  00FC  00FB  012B  012A  0010  000F  00F20           .WORD   15, 16, 298, 299, 251, 252, 172, 173, -
0138  00D5  0068  00E7  0004  0090  00A0  0073  0099  0085  00F34                   130, 134, 133, 153, 115, 160, 144, 4, -
                                          0084  0107  00B9  00F48                   231, 104, 213, 312, 185, 263, 132
                                                               00#  00F4E           .BYTE   0[18]
008C  0087  00AF  00AE  00FE  00FD  012D  012C  0012  0011  00F60           .WORD   17, 18, 300, 301, 253, 254, 174, 175, -
0139  00D6  0069  00E8  0005  0091  00A1  0074  009A  008B  00F74                   135, 140, 139, 154, 116, 161, 145, 5, -
```

```
                                   008A  0108  00BA  00F88                      232, 105, 214, 313, 186, 264, 138     ;
                                                 00#  00F8E           .BYTE     0[10]                                  ;
                             0112  00DF  00EA  0060  00F98           .WORD     96, 234, 223, 274                      ;
                                                 00#  00FA0           .BYTE     0[48]                                  ;
                                         007E  007F  00FD0           .WORD     127, 126                               ;
                                                 00#  00FD4           .BYTE     0[184]                                 ;
                                         0088  0089  0108C           .WORD     137, 136                               ;
                                                 00#  01090           .BYTE     0[10]                                  ;
                                         004E  004D  0109A           .WORD     77, 78                                 ;
                                                      0109E           .BLKB     2
        02  01  10  08  08  04  10  08  04  02  01  010A0 DATA_SIZE:
                                                                      .BYTE     1, 2, 4, 8, 16, 4, 8, 8, 16, 1, 2     ;
                                                      010AB           .BLKB     1
        03  02  1C  1B  0B  0A  1A  09  08  07  06  010AC DATA_TYPE:
                                                                      .BYTE     6, 7, 8, 9, 26, 10, 11, 27, 28, 2, 3  ;
                                                      010B7           .BLKB     1
                                                  02  010B8 P.AAB:    .BYTE     2                                      ;
                                                  52  010B9           .ASCII    \R\                                    ;
                                                  30  010BA           .ASCII    \0\                                    ;
                                              02  00  010BB           .BYTE     0, 2                                   ;
                                                  52  010BD           .ASCII    \R\                                    ;
                                                  31  010BE           .ASCII    \1\                                    ;
                                              02  00  010BF           .BYTE     0, 2                                   ;
                                                  52  010C1           .ASCII    \R\                                    ;
                                                  32  010C2           .ASCII    \2\                                    ;
                                              02  00  010C3           .BYTE     0, 2                                   ;
                                                  52  010C5           .ASCII    \R\                                    ;
                                                  33  010C6           .ASCII    \3\                                    ;
                                              02  00  010C7           .BYTE     0, 2                                   ;
                                                  52  010C9           .ASCII    \R\                                    ;
                                                  34  010CA           .ASCII    \4\                                    ;
                                              02  00  010CB           .BYTE     0, 2                                   ;
                                                  52  010CD           .ASCII    \R\                                    ;
                                                  35  010CE           .ASCII    \5\                                    ;
                                              02  00  010CF           .BYTE     0, 2                                   ;
                                                  52  010D1           .ASCII    \R\                                    ;
                                                  36  010D2           .ASCII    \6\                                    ;
                                              02  00  010D3           .BYTE     0, 2                                   ;
                                                  52  010D5           .ASCII    \R\                                    ;
                                                  37  010D6           .ASCII    \7\                                    ;
                                              02  00  010D7           .BYTE     0, 2                                   ;
                                                  52  010D9           .ASCII    \R\                                    ;
                                                  38  010DA           .ASCII    \8\                                    ;
                                              02  00  010DB           .BYTE     0, 2                                   ;
                                                  52  010DD           .ASCII    \R\                                    ;
                                                  39  010DE           .ASCII    \9\                                    ;
                                              03  00  010DF           .BYTE     0, 3                                   ;
                                                  52  010E1           .ASCII    \R\                                    ;
                                                  31  010E2           .ASCII    \1\                                    ;
                                                  30  010E3           .ASCII    \0\                                    ;
                                                  03  010E4           .BYTE     3                                      ;
                                                  52  010E5           .ASCII    \R\                                    ;
                                                  31  010E6           .ASCII    \1\                                    ;
                                                  31  010E7           .ASCII    \1\                                    ;
                                                  02  010E8           .BYTE     2                                      ;
                                                  41  010E9           .ASCII    \A\                                    ;
                                                  50  010EA           .ASCII    \P\                                    ;
```

```
                        02  00  010EB          .BYTE   0, 2
                            46  010ED          .ASCII  \F\
                            50  010EE          .ASCII  \P\
                        02  00  010EF          .BYTE   0, 2
                            53  010F1          .ASCII  \S\
                            50  010F2          .ASCII  \P\
                        02  00  010F3          .BYTE   0, 2
                            50  010F5          .ASCII  \P\
                            43  010F6          .ASCII  \C\
                        02  00  010F7          .BYTE   0, 2
                            3F  010F9          .ASCII  \?\
                            3F  010FA          .ASCII  \?\
                        02  00  010FB          .BYTE   0, 2
                            3F  010FD          .ASCII  \?\
                            3F  010FE          .ASCII  \?\
                        02  00  010FF          .BYTE   0, 2
                            49  01101          .ASCII  \I\
                            56  01102          .ASCII  \V\
                        02  00  01103          .BYTE   0, 2
                            44  01105          .ASCII  \D\
                            56  01106          .ASCII  \V\
                            00  01107          .BYTE   0
44  41  21  03  01108 P.AAC:   .ASCII  <3>\!AD\
43  41  21  03  0110C P.AAD:   .ASCII  <3>\!AC\
                    2C  01110 P.AAE:   .ASCII  \,\

                                       .PSECT  DBG$OWN,NOEXE,  PIC,2

                            00000 OP_BUFFER:
                                       .BLKB   16

                            DBG$OPCODE_NAME_TABLE==
                                               P.AAA
                            OPERAND_VALUE=     OP_BUFFER
                            REGISTER_NAME=     P.AAB
                            FORMAT_AD=         P.AAC
                            FORMAT_AC=         P.AAD
                            COMMA=             P.AAE
                                       .EXTRN  DBG$GB_RADIX, DBG$CONV_TEXT_VALUE
                                       .EXTRN  DBG$COVER_DX_DX
                                       .EXTRN  DBG$PRINT, DBG$PRINT_VALUE
                                       .EXTRN  DBG$PRINT_IDENTIFIER_PC
                                       .EXTRN  DBG$NEWLINE, DBG$POP_TEMPMEM
                                       .EXTRN  DBG$PUSH_TEMPMEM
                                       .EXTRN  DBG$IS_IT_ENTRY
                                       .EXTRN  DBG$MAKE_VAL_DESC
                                       .EXTRN  DBG$NPARSE_ADDRESS
                                       .EXTRN  DBG$NPARSE_EXPRESSION
                                       .EXTRN  DBG$PRIM_TO_VAL
                                       .EXTRN  SYS$UNWIND

                                       .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                                0000 00000 DECODE_HANDLER:
                                       .WORD   Save nothing                    ; 0793
                    50      04  AC  D0 00002          MOVL    SIG_ARGS, R0       ; 0800
00000920            8F      04  A0  D1 00006          CMPL    4(R0), #2336       ;
```

```
                              20    13 0000E         BEQL    2$                        ;
             51        0C     AC    D0 00010         MOVL    ENABLE_ARGS, R1           ; 0801
             06        04     B1    E9 00014         BLBC    @4(R1), 1$                ;
             50      0918     8F    3C 00018         MOVZWL  #2328, R0                 ;
                                    04 0001D         RET                               ;
             50        08     AC    D0 0001E 1$:     MOVL    MECH_ARGS, R0             ; 0803
       0C    A0        08     B1    D0 00022         MOVL    @8(R1), 12(R0)            ;
                              7E    7C 00027         CLRQ    -(SP)                     ; 0805
 00000000G   00               02    FB 00029         CALLS   #2, SYS$UNWIND            ;
             50               01    D0 00030 2$:     MOVL    #1, R0                    ; 0806
                                    04 00033         RET                               ; 0807

; Routine Size:  52 bytes,    Routine Base:  DBG$CODE + 0000


;  685          0808  2
;  686          0809  2     ENABLE Decode_Handler(Signal_Flag,Error_Value);
;  687          0810  2
;  688          0811  2     Signal_Flag = .Print_Flag;
;  689          0812  2     Error_Value = .Start_Address + 1;
```

```
;  691          0813  2     Pointer = .Start_Address;
;  692          0814  2
;  693          0815  2     !++
;  694          0816  2     ! Do we have an entry mask
;  695          0817  2     !--
;  696          0818  3     IF (IF ActualCount() GTR 2 THEN .Entry_Flag ELSE DBG$Is_It_Entry(.Start_Address))
;  697          0819  2     THEN
;  698          0820  3         BEGIN
;  699          0821  3         Error_Value = .Error_Value + 1;
;  700          0822  3         Fetch_Instruction(Pointer,context_wu);
;  701          0823  3         IF .Print_Flag THEN
;  702          0824  4             BEGIN
;  703          0825  4             LOCAL Delimiter,Mask_Bits;
;  704          0826  4
;  705          0827  4             Mask_Bits = .Operand_Value<0,12,0>+(.Operand_Value<12,4,0>)^16;
;  706          0828  4
;  707          0829  4             DBG$Print(UPLIT BYTE (%ASCIC 'entry mask ^M'));
;  708          0830  4
;  709          0831  4             Delimiter = %C'<';
;  710          0832  4
;  711          0833  4             IF .Mask_Bits EQL 0 THEN DBG$Print(Format_AD,1,Delimiter) ELSE
;  712          0834  4               INCR Index FROM 0 TO 19 DO IF .Mask_Bits<.Index,1,0> THEN
;  713          0835  5                 BEGIN
;  714          0836  5                 DBG$Print(Format_AD,1,Delimiter);
;  715          0837  5                 DBG$Print(Format_AC,Register_Name[.Index]);
;  716          0838  5                 Delimiter = %C',';
;  717          0839  4                 END;
;  718          0840  4
;  719          0841  4             DBG$Print(Format_AD,1,UPLIT BYTE('>'));
;  720          0842  4
;  721          0843  4             IF .Operand_Value<12,2,0> NEQ 0 THEN
;  722          0844  5                 BEGIN
;  723          0845  5                 DBG$Newline();
;  724          0846  5                 SIGNAL(dbg$_entrymask);
;  725          0847  4                 END;
;  726          0848  3             END;
;  727          0849  3         END
```

```
;  729          0850  2     ELSE                    ! Not an entry mask - decode actual instruction
;  730          0851  2         BEGIN
;  731          0852  2         LOCAL
;  732          0853  2             Opcode,
;  733          0854  2             Opcode_Index,
;  734          0855  2             Opcode_Entry : REF BLOCK[10,BYTE] FIELD(Opcode_Entry_Fields),
;  735          0856  2             Delimiter,
;  736          0857  2             S* .e;
;  737          0858  2
;  738          0859  2         Fetch_Instruction(Pointer,context_bu);
;  739          0860  2         Opcode = .Op_Buffer[u_byte];
;  740          0861  2         Opcode_Index = .DBG$Opcode_Kind_Table[.Opcode];
;  741          0862  3         IF (.Opcode_Index EQL 0) THEN
;  742          0863  4             BEGIN
;  743          0864  4             IF (.Opcode GEQU %X'FD') THEN
;  744          0865  5                 BEGIN
;  745          0866  5                 Fetch_Instruction(Pointer,context_bu);
;  746          0867  7                 IF ((.Opcode EQLU %X'FD') AND (.Op_Buffer[u_byte]  LSSU %X'FD'))
;  747          0868  6                  OR ((.Opcode EQLU %X'FF') AND (.Op_Buffer[u_byte]  GEQU %X'FD')))
;  748          0869  5                   THEN
;  749          0870  5                     Opcode_Index = .DBG$Opcode_Kind_Table[.Op_Buffer[u_byte]+%X'100'];
;  750          0871  4                 END;
;  751          0872  4             IF (.Opcode_Index EQL 0) THEN SIGNAL(dbg$_noinstran,1,.Start_Address);
;  752          0873  3             END;
;  753          0874  3
;  754          0875  3         Opcode_Entry = DBG$Opcode_Name_Table[.Opcode_Index,offset];
;  755          0876  3
;  756          0877  3         IF .Print_Flag THEN DBG$Print(Format_AD,6,Opcode_Entry[op_name]);
;  757          0878  3         Delimiter = %C' ';
```

```
;  759          0879  3          State = .Opcode_Entry[op_kind];
;  760          0880  3
;  761          0881  3          WHILE (.State NEQ simple_0_operand) DO
;  762          0882  4              BEGIN
;  763          0883  4              IF .Print_Flag THEN DBG$Print(Format_AD,1,Delimiter);
;  764          0884  4              Delimiter = %C',';
;  765          0885  4
;  766          0886  4              CASE .State FROM simple_1_operand TO maximum_state OF
;  767          0887  4                  SET
;  768          0888  4                  [simple_1_operand,
;  769          0889  4                   simple_2_operand,
;  770          0890  4                   simple_3_operand]:
;  771          0891  5                      BEGIN
;  772          0892  5                      Fetch_Operand(Pointer,.Opcode_Entry[op_type_one],.Print_Flag,0);
;  773          0893  5                      State = .State - 1;
;  774          0894  4                      END;
;  775          0895  4
;  776          0896  4                  [branch_0_operand]:
;  777          0897  5                      BEGIN
;  778          0898  5                      Fetch_Instruction(Pointer,.Opcode_Entry[op_type_one]);
;  779          0899  5                      IF .Print_Flag THEN
;  780          0900  5                          Print_Address(.Operand_Value + .Pointer);
;  781          0901  5                      EXITLOOP;
;  782          0902  4                      END;
;  783          0903  4
;  784          0904  4                  [branch_1_operand,
;  785          0905  4                   branch_2_operand,
;  786          0906  4                   branch_3_operand]:
;  787          0907  5                      BEGIN
;  788          0908  5                      Fetch_Operand(Pointer,.Opcode_Entry[op_type_two],.Print_Flag,0);
;  789          0909  5                      State = .State - 1;
;  790          0910  4                      END;
```

```
;  792          0911  4                    [convert_datatype,
;  793          0912  4 ! *****             evaluate_address,
;  794          0913  4 ! *****             simple_bit_field,
;  795          0914  4 ! *****             routine_dispatch,
;  796          0915  4 ! *****             polynomial_value,
;  797          0916  4 ! *****             probe_for_access,
;  798          0917  4                     string_3_operand]:
;  799          0918  5                         BEGIN
;  800          0919  5                         Fetch_Operand(Pointer,.Opcode_Entry[op_type_two],.Print_Flag,0);
;  801          0920  5                         State = Simple_1_Operand;
;  802          0921  4                         END;
;  803          0922  4
;  804          0923  4                     [string_4_operand]:
;  805          0924  5                         BEGIN
;  806          0925  5                         Fetch_Operand(Pointer,.Opcode_Entry[op_type_two],.Print_Flag,0);
;  807          0926  5                         State = Simple_2_Operand;
;  808          0927  4                         END;
;  809          0928  4
;  810          0929  4                     [string_5_operand,
;  811          0930  4                      string_6_operand]:
;  812          0931  5                         BEGIN
;  813          0932  5                         Fetch_Operand(Pointer,.Opcode_Entry[op_type_two],.Print_Flag,0);
;  814          0933  5                         DECR count FROM .State TO string_5_operand DO
;  815          0934  6                             BEGIN
;  816          0935  6                             IF .Print_Flag THEN DBG$Print(Format_AD,1,Delimiter);
;  817          0936  6                             Fetch_Operand(Pointer,context_b,.Print_Flag,0);
;  818          0937  5                             END;
;  819          0938  5                         State = Simple_1_Operand;
;  820          0939  4                         END;
;  821          0940  4
;  822          0941  4                     [trailing_operand]:
;  823          0942  5                         BEGIN
;  824          0943  5                         Fetch_Instruction(Pointer,.Opcode_Entry[op_type_one]);
;  825          0944  5                         IF .Print_Flag THEN Print_Operand(.Opcode_Entry[op_type_one]);
;  826          0945  5                         EXITLOOP;
;  827          0946  4                         END;
```

```
;  829          0947  4                [complex_SHIFT]:
;  830          0948  5                    BEGIN
;  831          0949  5                    Fetch_Operand(Pointer,context_b,.Print_Flag,0);
;  832          0950  5                    State = Simple_2_Operand;
;  833          0951  4                    END;
;  834          0952  4
;  835          0953  4                [complex_EMOD]:
;  836          0954  5                    BEGIN
;  837          0955  5                    Fetch_Operand(Pointer,.Opcode_Entry[op_type_one],.Print_Flag,0);
;  838          0956  5                    IF .Print_Flag THEN DBG$Print(Format_AD,1,Delimiter);
;  839          0957  5                    Fetch_Operand(Pointer,.Opcode_Entry[op_type_two],.Print_Flag,0);
;  840          0958  5                    State = Simple_3_Operand;
;  841          0959  4                    END;
;  842          0960  4
;  843          0961  4                [complex_CRC]:
;  844          0962  5                    BEGIN
;  845          0963  5                    Fetch_Operand(Pointer,context_b,.Print_Flag,0);
;  846          0964  5                    IF .Print_Flag THEN DBG$Print(Format_AD,1,Delimiter);
;  847          0965  5                    Fetch_Operand(Pointer,context_l,.Print_Flag,0);
;  848          0966  5                    IF .Print_Flag THEN DBG$Print(Format_AD,1,Delimiter);
;  849          0967  5                    Fetch_Operand(Pointer,context_t,.Print_Flag,0);
;  850          0968  5                    EXITLOOP;
;  851          0969  4                    END;
;  852          0970  4
;  853          0971  4                [complex_EMUL]:
;  854          0972  5                    BEGIN
;  855          0973  5                    Fetch_Operand(Pointer,context_l,.Print_Flag,0);
;  856          0974  5                    IF .Print_Flag THEN DBG$Print(Format_AD,1,Delimiter);
;  857          0975  5                    Fetch_Operand(Pointer,context_l,.Print_Flag,0);
;  858          0976  5                    IF .Print_Flag THEN DBG$Print(Format_AD,1,Delimiter);
;  859          0977  5                    Fetch_Operand(Pointer,context_l,.Print_Flag,0);
;  860          0978  5                    IF .Print_Flag THEN DBG$Print(Format_AD,1,Delimiter);
;  861          0979  5                    Fetch_Operand(Pointer,context_q,.Print_Flag,0);
;  862          0980  5                    EXITLOOP;
;  863          0981  4                    END;
;  864          0982  4
;  865          0983  4                [complex_EDIV]:
;  866          0984  5                    BEGIN
;  867          0985  5                    Fetch_Operand(Pointer,context_l,.Print_Flag,0);
;  868          0986  5                    IF .Print_Flag THEN DBG$Print(Format_AD,1,Delimiter);
;  869          0987  5                    Fetch_Operand(Pointer,context_q,.Print_Flag,0);
;  870          0988  5                    IF .Print_Flag THEN DBG$Print(Format_AD,1,Delimiter);
;  871          0989  5                    Fetch_Operand(Pointer,context_l,.Print_Flag,0);
;  872          0990  5                    IF .Print_Flag THEN DBG$Print(Format_AD,1,Delimiter);
;  873          0991  5                    Fetch_Operand(Pointer,context_l,.Print_Flag,0);
;  874          0992  5                    EXITLOOP;
;  875          0993  4                    END;
```

```
;  877          0994  4          [complex_INDEX]:
;  878          0995  5              BEGIN
;  879          0996  5              DECR count FROM 5 TO 0 DO
;  880          0997  6                  BEGIN
;  881          0998  6                  Fetch_Operand(Pointer,context_l,.Print_Flag,0);
;  882          0999  6                  IF .Print_Flag AND (.count NEQ 0) THEN DBG$Print(Format_AD,1,Delimiter);
;  883          1000  5                  END;
;  884          1001  5              EXITLOOP;
;  885          1002  4              END;
;  886          1003  4
;  887          1004  4          [complex_CASE]:
;  888          1005  5              BEGIN
;  889          1006  5              LOCAL                                                                   !
;  890          1007  5                  Limit;                                                              ! The number of displacements
;  891          1008  5              Fetch_Operand(Pointer,.Opcode_Entry[op_type_one],.Print_Flag,0);
;  892          1009  5              IF .Print_Flag THEN DBG$Print(Format_AD,1,Delimiter);
;  893          1010  5              Fetch_Operand(Pointer,.Opcode_Entry[op_type_one],.Print_Flag,0);
;  894          1011  5              IF .Print_Flag THEN DBG$Print(Format_AD,1,Delimiter);
;  895          1012  5
;  896          1013  5              !++                                                                     !
;  897          1014  5              ! Fetch the limit                                                       !
;  898          1015  5              !--                                                                     !
;  899          1016  5              Fetch_Operand(Pointer,.Opcode_Entry[op_type_one],.Print_Flag,0);
;  900          1017  5
;  901          1018  5              !++                                                                     !
;  902          1019  5              ! Get the limit in of the appropriate length                            !
;  903          1020  5              !--                                                                     !
;  904          1021  6              Limit = (SELECTONE .Opcode_Entry[ Op_type_one ] OF                      ! What is the operand length?
;  905          1022  6                      SET                                                             !
;  906          1023  6                      [context_b]:        .Op_buffer[u_byte];                         ! Byte
;  907          1024  6                      [context_w]:        .Op_buffer[u_word];                         ! Word
;  908          1025  6                      [context_l]:        .Op_buffer[u_long];                         ! Long
;  909          1026  5                      TES);                                                           !
;  910          1027  5
;  911          1028  5              IF NOT .Print_Flag
;  912          1029  5              THEN
;  913          1030  6                  Pointer = .Pointer+2*(.Limit+1)                                     !
;  914          1031  5              ELSE
;  915          1032  6                  BEGIN
;  916          1033  6                  LOCAL start;
;  917          1034  6                  start = .Pointer;
;  918          1035  6                  DECR count FROM .Limit TO 0 DO
;  919          1036  7                      BEGIN
;  920          1037  7                      BIND                                                            !
;  921          1038  7                          CASE_Offset = Op_buffer[u_word] : SIGNED;                    ! Convert the word to signed
;  922          1039  7                      $ABORT_ON_CONTROL_Y;                                             ! Its ok the ^Y out of the ca
;  923          1040  7                      Fetch_Instruction(Pointer,context_w);
;  924          1041  7                      DBG$NewLine();
;  925          1042  7                      DBG$Print(Format_AD,2,UPLIT BYTE('          '));
;  926          1043  7                      Print_Address(.CASE_Offset + .Start);                            ! Print the address
;  927          1044  6                      END;
;  928          1045  5                  END;
;  929          1046  5              EXITLOOP;
;  930          1047  4              END;
;  931          1048  4
;  932          1049  4          [complex_ASHP]:                                                                 ! ASHP didn't fit an
;  933          1050  5              BEGIN                                                                       !
```

```
;  934          1051  5                    Fetch_Operand(Pointer,.Opcode_Entry[op_type_two],.Print_Flag,0);
;  935          1052  5                    IF .Print_Flag THEN DBG$Print(Format_AD,1,Delimiter);
;  936          1053  5                    Fetch_Operand(Pointer,.Opcode_Entry[op_type_one],.Print_Flag,0);
;  937          1054  5                    IF .Print_Flag THEN DBG$Print(Format_AD,1,Delimiter);
;  938          1055  5                    Fetch_Operand(Pointer,.Opcode_Entry[op_type_two],.Print_Flag,0);
;  939          1056  5                    IF .Print_Flag THEN DBG$Print(Format_AD,1,Delimiter);
;  940          1057  5                    Fetch_Operand(Pointer,.Opcode_Entry[op_type_one],.Print_Flag,0);
;  941          1058  5                    EXITLOOP;
;  942          1059  4                    END;
;  943          1060  4
;  944          1061  4                [INRANGE,OUTRANGE]:
;  945          1062  4                    $DBG_ERROR('DBG$Encode_Decode - bad opcode table entry');
;  946          1063  4                TES;
;  947          1064  3            END;
;  948          1065  2        END;
;  949          1066  2
;  950          1067  2    RETURN .Pointer;
;  951          1068  1    END;


                                                                .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

    4D  5E  20  6B  73  61  6D  20  79  72  74  6E  65  0D  01111 P.AAF:  .ASCII  <13>\entry mask ^M\                          ;
                                                        3E  0111F P.AAG:  .ASCII  \>\                                       ;
                                                    09  09  01120 P.AAH:  .ASCII  <9><9>                                    ;
63  65  44  5F  65  64  6F  63  6E  45  24  47  42  44  2A  01122 P.AAI:  .ASCII  \*DBG$Encode_Decode - bad opcode table en\  ;
64  6F  63  70  6F  20  64  61  62  20  2D  20  65  64  6F  01131                                                             ;
                    6E  65  20  65  6C  62  61  74  20  65  01140                                                           ;
                                                79  72  74  0114A          .ASCII  \try\                                    ;

                                                CASE_OFFSET=        OP_BUFFER
                                                           .EXTRN   DBG$GV_CONTROL

                                                           .PSECT   DBG$CODE,NOWRT,  SHR,  PIC,0

                                                    0FFC 00000             .ENTRY   DBG$INS_DECODE, Save R2,R3,R4,R5,R6,R7,R8,- ; 0785
                                                                                    R9,R10,R11                                  ;
                                5B 00000000G    00  9E 00002               MOVAB    LIB$SIGNAL, R11                           ;
                                5A      0000V   CF  9E 00009               MOVAB    FETCH_OPERAND, R10                        ;
                                59 00000000'    EF  9E 0000E               MOVAB    OPERAND_VALUE, R9                         ;
                                58 00000000G    00  9E 00015               MOVAB    DBG$PRINT, R8                             ;
                                57 00000000'    EF  9E 0001C               MOVAB    FORMAT_AD, R7                             ;
                                5E                  14  C2 00023           SUBL2    #20, SP                                   ;
                                             0C     AE  7C 00026           CLRQ     ERROR_VALUE                               ; 0786
                                6D         04A1     CF  DE 00029           MOVAL    72$, (FP)                                 ;
                                56           08     AC  D0 0002E           MOVL     PRINT_FLAG, R6                            ; 0811
                                      10     AE          56  D0 00032       MOVL     R6, SIGNAL_FLAG                           ;
                                0C    AE     04     AC  01  C1 00036       ADDL3    #1, START_ADDRESS, ERROR_VALUE            ; 0812
                                      08     AE     04   AC  D0 0003C       MOVL     START_ADDRESS, POINTER                    ; 0813
                                      02           6C  91 00041           CMPB     (AP), #2                                  ; 0818
                                                   07  1B 00044           BLEQU    1$                                        ;
                                      10     0C   AC  E8 00046           BLBS     ENTRY_FLAG, 2$                            ;
                                                 008A  31 00049           BRW      8$                                        ;
                                             04   AC  DD 0004D 1$:        PUSHL    START_ADDRESS                             ;
                                00000000G    00      01  FB 00050       CALLS    #1, DBG$IS_IT_ENTRY                       ;
                                      7D           50  E9 00057           BLBC     R0, 8$                                    ;
```

```
                                                0C   AE   D6 0005A 2$:     INCL    ERROR_VALUE                                    ; 0821
                                                0A   DD   0005D           PUSHL   #10                                            ; 0822
                                                0C   AE   9F 0005F        PUSHAB  POINTER                                        ;
                                      0000V     CF        02 FB 00062     CALLS   #2, FETCH_INSTRUCTION                          ;
                                                6A        56 E9 00067     BLBC    R6, 7$                                         ; 0823
                                50    01   A9             04 EF 0006A     EXTZV   #4, #4, OPERAND_VALUE+1, R0                    ; 0827
                                                50        10 78 00070     ASHL    #16, R0, R0                                    ;
                                53          69             00 EF 00074     EXTZV   #0, #12, OPERAND_VALUE, MASK_BITS              ;
                                                53        50 C0 00079     ADDL2   R0, MASK_BITS                                  ;
                                                09   A7   9F 0007C        PUSHAB  P.AAF                                          ; 0829
                                                68        01 FB 0007F     CALLS   #1, DBG$PRINT                                  ;
                                                6E        3C D0 00082     MOVL    #60, DELIMITER                                 ; 0831
                                                          53 D5 00085     TSTL    MASK_BITS                                      ; 0833
                                                          0B 12 00087     BNEQ    3$                                             ;
                                                          5E DD 00089     PUSHL   SP                                             ;
                                                          01 DD 0008B     PUSHL   #1                                             ;
                                                          57 DD 0008D     PUSHL   R7                                             ;
                                                68        03 FB 0008F     CALLS   #3, DBG$PRINT                                  ;
                                                          20 11 00092     BRB     6$                                             ;
                                                          52 D4 00094 3$: CLRL    INDEX                                          ; 0834
                                          16              52 E1 00096 4$: BBC     INDEX, MASK_BITS, 5$                           ;
                                                          5E DD 0009A     PUSHL   SP                                             ; 0836
                                                          01 DD 0009C     PUSHL   #1                                             ;
                                                          57 DD 0009E     PUSHL   R7                                             ;
                                                68        03 FB 000A0     CALLS   #3, DBG$PRINT                                  ;
                                                B0 A742   DF 000A3        PUSHAL  REGISTER_NAME[INDEX]                           ; 0837
                                                04   A7   9F 000A7        PUSHAB  FORMAT_AC                                      ;
                                                68        02 FB 000AA     CALLS   #2, DBG$PRINT                                  ;
                                                6E        2C D0 000AD     MOVL    #44, DELIMITER                                 ; 0838
                                          E2    52        13 F3 000B0 5$: AOBLEQ  #19, INDEX, 4$                                 ; 0834
                                                17   A7   9F 000B4 6$:    PUSHAB  P.AAG                                          ; 0841
                                                          01 DD 000B7     PUSHL   #1                                             ;
                                                          57 DD 000B9     PUSHL   R7                                             ;
                                                68        03 FB 000BB     CALLS   #3, DBG$PRINT                                  ;
                                                30   01   A9 93 000BE     BITB    OPERAND_VALUE+1, #48                           ; 0843
                                                          10 13 000C2     BEQL    7$                                             ;
                                      00000000G 00        00 FB 000C4     CALLS   #0, DBG$NEWLINE                                ; 0845
                                                00028FA3  8F DD 000CB     PUSHL   #167843                                        ; 0846
                                                68        01 FB 000D1     CALLS   #1, LIB$SIGNAL                                 ;
                                                     03F2 31 000D4 7$:    BRW     71$                                            ; 0818
                                                          09 DD 000D7 8$:    PUSHL   #9                                          ; 0859
                                                     0C   AE 9F 000D9        PUSHAB  POINTER                                        ;
                                           0000V    CF        02 FB 000DC     CALLS   #2, FETCH_INSTRUCTION                          ;
                                                53          69 9A 000E1     MOVZBL  OP_BUFFER, OPCODE                              ; 0860
                                                52     FB98 C743 3C 000E4     MOVZWL  DBG$OPCODE_KIND_TABLE[OPCODE], OPCODE_INDEX    ; 0861
                                                          4C 12 000EA     BNEQ    12$                                            ; 0862
                                      000000FD  8F        53 D1 000EC     CMPL    OPCODE, #253                                   ; 0864
                                                          31 1F 000F3     BLSSU   11$                                            ;
                                                          09 DD 000F5     PUSHL   #9                                             ; 0866
                                                     0C   AE 9F 000F7        PUSHAB  POINTER                                        ;
                                           0000V    CF        02 FB 000FA     CALLS   #2, FETCH_INSTRUCTION                          ;
                                      000000FD  8F        53 D1 000FF     CMPL    OPCODE, #253                                   ; 0867
                                                          06 12 00106     BNEQ    9$                                             ;
                                            FD    8F        69 91 00108     CMPB    OP_BUFFER, #253                                ;
                                                          0F 1F 0010C     BLSSU   10$                                            ;
                                      000000FF  8F        53 D1 0010E 9$: CMPL    OPCODE, #255                                   ; 0868
                                                          0F 12 00115     BNEQ    11$                                            ;
```

```
                                FD   8F            69  91 00117          CMPB    OP_BUFFER, #253                                 ;
                                                  09  1F 0011B          BLSSU   11$                                             ;
                                     50           69  9A 0011D 10$:     MOVZBL  OP_BUFFER, R0                                   ; 0870
                                     52     FD98  C740  3C 00120          MOVZWL  DBG$OPCODE_KIND_TABLE+512[R0], OPCODE_INDEX      ;
                                                  52  D5 00126 11$:     TSTL    OPCODE_INDEX                                    ; 0872
                                                  0E  12 00128          BNEQ    12$                                             ;
                                        04        AC  DD 0012A          PUSHL   START_ADDRESS                                   ;
                                                  01  DD 0012D          PUSHL   #1                                              ;
                                   000281C8       8F  DD 0012F          PUSHL   #164296                                         ;
                                     6B           03  FB 00135          CALLS   #3, LIB$SIGNAL                                  ;
                                     52           0A  C4 00138 12$:     MULL2   #10, R2                                         ; 0875
                                     54     EEF8  C742  9E 0013B          MOVAB   DBG$OPCODE_NAME_TABLE[R2], OPCODE_ENTRY          ;
                                     09           56  E9 00141          BLBC    R6, 13$                                         ; 0877
                                                  54  DD 00144          PUSHL   OPCODE_ENTRY                                    ;
                                                  06  DD 00146          PUSHL   #6                                              ;
                                                  57  DD 00148          PUSHL   R7                                              ;
                                     68           03  FB 0014A          CALLS   #3, DBG$PRINT                                   ;
                                04   AE           09  D0 0014D 13$:     MOVL    #9, DELIMITER                                   ; 0878
                                     52     08    A4  9A 00151          MOVZBL  8(OPCODE_ENTRY), STATE                          ; 0879
                                                  52  D5 00155 14$:     TSTL    STATE                                           ; 0881
                                                  7A  13 00157          BEQL    20$                                             ;
                                     0A           56  E9 00159          BLBC    R6, 15$                                         ; 0883
                                        04        AE  9F 0015C          PUSHAB  DELIMITER                                       ;
                                                  01  DD 0015F          PUSHL   #1                                              ;
                                                  57  DD 00161          PUSHL   R7                                              ;
                                     68           03  FB 00163          CALLS   #3, DBG$PRINT                                   ;
                                04   AE           2C  D0 00166 15$:     MOVL    #44, DELIMITER                                  ; 0884
                                           16     01  CF 0016A          CASEL   STATE, #1, #22                                  ; 0886
                    004A          003E      003E      003E      0016E 16$:     .WORD   18$-16$,-                                       ;
                    007C          0068      0068      0068      00176          .WORD   18$-16$,-                                       ;
                    009A          008E      007C      00D4      0017E          .WORD   18$-16$,-                                       ;
                    00F3          002E      002E      009A      00186          .WORD   19$-16$,-                                       ;
                    0171          0104      01C4      023D      0018E          .WORD   21$-16$,-                                       ;
                                  02F4      0137      0218      00196          .WORD   21$-16$,-                                       ;
                                                                                .WORD   21$-16$,-                                       ;
                                                                                .WORD   23$-16$,-                                       ;
                                                                                .WORD   31$-16$,-                                       ;
                                                                                .WORD   23$-16$,-                                       ;
                                                                                .WORD   24$-16$,-                                       ;
                                                                                .WORD   25$-16$,-                                       ;
                                                                                .WORD   25$-16$,-                                       ;
                                                                                .WORD   17$-16$,-                                       ;
                                                                                .WORD   17$-16$,-                                       ;
                                                                                .WORD   32$-16$,-                                       ;
                                                                                .WORD   53$-16$,-                                       ;
                                                                                .WORD   45$-16$,-                                       ;
                                                                                .WORD   34$-16$,-                                       ;
                                                                                .WORD   40$-16$,-                                       ;
                                                                                .WORD   50$-16$,-                                       ;
                                                                                .WORD   37$-16$,-                                       ;
                                                                                .WORD   66$-16$                                         ;
                                        1A    A7  9F 0019C 17$:     PUSHAB  P.AAI                                           ; 1062
                                                  01  DD 0019F          PUSHL   #1                                              ;
                                   00028362       8F  DD 001A1          PUSHL   #164706                                         ;
                                     6B           03  FB 001A7          CALLS   #3, LIB$SIGNAL                                  ;
                                                  A9  11 001AA          BRB     14$                                             ;
                                                  7E  D4 001AC 18$:     CLRL    -(SP)                                           ; 0892
```

```
                                                    56  DD 001AE          PUSHL   R6
                7E      09      A4              04  00  EF 001B0          EXTZV   #0, #4, 9(OPCODE_ENTRY), -(SP)
                                                    28  11 001B6          BRB     22$
                7E      09      A4              04  00  EF 001B8 19$:     EXTZV   #0, #4, 9(OPCODE_ENTRY), -(SP)                ; 0898
                                                0C  AE  9F 001BE          PUSHAB  POINTER
                                        0000V   CF  02  FB 001C1          CALLS   #2, FETCH_INSTRUCTION
                                                0A  56  E9 001C6          BLBC    R6, 20$                                      ; 0899
                        7E                      69  08  AE  C1 001C9      ADDL3   POINTER, OPERAND_VALUE, -(SP)                ; 0900
                                        0000V   CF  01  FB 001CE          CALLS   #1, PRINT_ADDRESS
                                            02F3  31 001D3 20$:           BRW     71$                                          ; 0897
                                               7E  D4 001D6 21$:          CLRL    -(SP)                                        ; 0908
                                               56  DD 001D8               PUSHL   R6
                7E      09      A4              04  04  EF 001DA          EXTZV   #4, #4, 9(OPCODE_ENTRY), -(SP)
                                                14  AE  9F 001E0 22$:     PUSHAB  POINTER
                                                6A  04  FB 001E3          CALLS   #4, FETCH_OPERAND
                                                    52  D7 001E6          DECL    STATE                                        ; 0909
                                                    56  11 001E8          BRB     30$                                          ; 0886
                                                    7E  D4 001EA 23$:     CLRL    -(SP)                                        ; 0919
                                                    56  DD 001EC          PUSHL   R6
                7E      09      A4              04  04  EF 001EE          EXTZV   #4, #4, 9(OPCODE_ENTRY), -(SP)
                                                14  AE  9F 001F4          PUSHAB  POINTER
                                                6A  04  FB 001F7          CALLS   #4, FETCH_OPERAND
                                                    41  11 001FA          BRB     29$                                          ; 0920
                                                    7E  D4 001FC 24$:     CLRL    -(SP)                                        ; 0925
                                                    56  DD 001FE          PUSHL   R6
                7E      09      A4              04  04  EF 00200          EXTZV   #4, #4, 9(OPCODE_ENTRY), -(SP)
                                                    5F  11 00206          BRB     33$
                                                    7E  D4 00208 25$:     CLRL    -(SP)                                        ; 0932
                                                    56  DD 0020A          PUSHL   R6
                7E      09      A4              04  04  EF 0020C          EXTZV   #4, #4, 9(OPCODE_ENTRY), -(SP)
                                                14  AE  9F 00212          PUSHAB  POINTER
                                                6A  04  FB 00215          CALLS   #4, FETCH_OPERAND
                                                53  52  D0 00218          MOVL    STATE, COUNT                                 ; 0933
                                                    1B  11 0021B          BRB     28$
                                                0A  56  E9 0021D 26$:     BLBC    R6, 27$                                      ; 0935
                                                04  AE  9F 00220          PUSHAB  DELIMITER
                                                    01  DD 00223          PUSHL   #1
                                                    57  DD 00225          PUSHL   R7
                                                68  03  FB 00227          CALLS   #3, DBG$PRINT
                                                    7E  D4 0022A 27$:     CLRL    -(SP)                                        ; 0936
                                                    56  DD 0022C          PUSHL   R6
                                                    7E  D4 0022E          CLRL    -(SP)
                                                14  AE  9F 00230          PUSHAB  POINTER
                                                6A  04  FB 00233          CALLS   #4, FETCH_OPERAND
                                                    53  D7 00236          DECL    COUNT                                        ; 0933
                                                0C  53  D1 00238 28$      CMPL    COUNT, #12
                                                    E0  18 0023B          BGEQ    26$
                                                52  01  D0 0023D 29$:     MOVL    #1, STATE                                    ; 0938
                                                    60  11 00240 30$:     BRB     36$                                          ; 0886
                7E      09      A4              04  00  EF 00242 31$:     EXTZV   #0, #4, 9(OPCODE_ENTRY), -(SP)               ; 0943
                                                0C  AE  9F 00248          PUSHAB  POINTER
                                        0000V   CF  02  FB 0024B          CALLS   #2, FETCH_INSTRUCTION
                                                80  56  E9 00250          BLBC    R6, 20$                                      ; 0944
                7E      09      A4              04  00  EF 00253          EXTZV   #0, #4, 9(OPCODE_ENTRY), -(SP)
                                        0000V   CF  01  FB 00259          CALLS   #1, PRINT_OPERAND
                                            0268  31 0025E              BRW     71$                                          ; 0942
                                               7E  D4 00261 32$:          CLRL    -(SP)                                        ; 0949
```

```
                                      56  DD 00263          PUSHL   R6
                                      7E  D4 00265          CLRL    -(SP)
                                 14   AE  9F 00267  33$:    PUSHAB  POINTER
                          6A          04  FB 0026A          CALLS   #4, FETCH_OPERAND
                          52          02  D0 0026D          MOVL    #2, STATE                              0950
                                      30  11 00270          BRB     36$                                    0886
                                      7E  D4 00272  34$:    CLRL    -(SP)                                  0955
                                      56  DD 00274          PUSHL   R6
        7E        09    A4        04  00  EF 00276          EXTZV   #0, #4, 9(OPCODE_ENTRY), -(SP)
                                 14   AE  9F 0027C          PUSHAB  POINTER
                          6A          04  FB 0027F          CALLS   #4, FETCH_OPERAND
                          0A          56  E9 00282          BLBC    R6, 35$                                0956
                                 04   AE  9F 00285          PUSHAB  DELIMITER
                                      01  DD 00288          PUSHL   #1
                                      57  DD 0028A          PUSHL   R7
                          68          03  FB 0028C          CALLS   #3, DBG$PRINT
                                      7E  D4 0028F  35$:    CLRL    -(SP)                                  0957
                                      56  DD 00291          PUSHL   R6
        7E        09    A4        04  04  EF 00293          EXTZV   #4, #4, 9(OPCODE_ENTRY), -(SP)
                                 14   AE  9F 00299          PUSHAB  POINTER
                          6A          04  FB 0029C          CALLS   #4, FETCH_OPERAND
                          52          03  D0 0029F          MOVL    #3, STATE                              0958
                                    FEB0  31 002A2  36$:    BRW     14$                                    0886
                                      7E  D4 002A5  37$:    CLRL    -(SP)                                  0963
                                      56  DD 002A7          PUSHL   R6
                                      7E  D4 002A9          CLRL    -(SP)
                                 14   AE  9F 002AB          PUSHAB  POINTER
                          6A          04  FB 002AE          CALLS   #4, FETCH_OPERAND
                          0A          56  E9 002B1          BLBC    R6, 38$                                0964
                                 04   AE  9F 002B4          PUSHAB  DELIMITER
                                      01  DD 002B7          PUSHL   #1
                                      57  DD 002B9          PUSHL   R7
                          68          03  FB 002BB          CALLS   #3, DBG$PRINT
                                      7E  D4 002BE  38$:    CLRL    -(SP)                                  0965
                                      56  DD 002C0          PUSHL   R6
                                      02  DD 002C2          PUSHL   #2
                                 14   AE  9F 002C4          PUSHAB  POINTER
                          6A          04  FB 002C7          CALLS   #4, FETCH_OPERAND
                          0A          56  E9 002CA          BLBC    R6, 39$                                0966
                                 04   AE  9F 002CD          PUSHAB  DELIMITER
                                      01  DD 002D0          PUSHL   #1
                                      57  DD 002D2          PUSHL   R7
                          68          03  FB 002D4          CALLS   #3, DBG$PRINT
                                      7E  D4 002D7  39$:    CLRL    -(SP)                                  0967
                                      56  DD 002D9          PUSHL   R6
                                      0B  DD 002DB          PUSHL   #11
                                      51  11 002DD          BRB     44$
                                      7E  D4 002DF  40$:    CLRL    -(SP)                                  0973
                                      56  DD 002E1          PUSHL   R6
                                      02  DD 002E3          PUSHL   #2
                                 14   AE  9F 002E5          PUSHAB  POINTER
                          6A          04  FB 002E8          CALLS   #4, FETCH_OPERAND
                          0A          56  E9 002EB          BLBC    R6, 41$                                0974
                                 04   AE  9F 002EE          PUSHAB  DELIMITER
                                      01  DD 002F1          PUSHL   #1
                                      57  DD 002F3          PUSHL   R7
                          68          03  FB 002F5          CALLS   #3, DBG$PRINT
```

```
                        7E  D4 002F8 41$:    CLRL    -(SP)                                   ;0975
                        56  DD 002FA         PUSHL   R6
                        02  DD 002FC         PUSHL   #2
                    14  AE  9F 002FE         PUSHAB  POINTER
                6A      04  FB 00301         CALLS   #4, FETCH_OPERAND
                0A      56  E9 00304         BLBC    R6, 42$                                 ;0976
                    04  AE  9F 00307         PUSHAB  DELIMITER
                        01  DD 0030A         PUSHL   #1
                        57  DD 0030C         PUSHL   R7
                68      03  FB 0030E         CALLS   #3, DBG$PRINT
                        7E  D4 00311 42$:    CLRL    -(SP)                                   ;0977
                        56  DD 00313         PUSHL   R6
                        02  DD 00315         PUSHL   #2
                    14  AE  9F 00317         PUSHAB  POINTER
                6A      04  FB 0031A         CALLS   #4, FETCH_OPERAND
                0A      56  E9 0031D         BLBC    R6, 43$                                 ;0978
                    04  AE  9F 00320         PUSHAB  DELIMITER
                        01  DD 00323         PUSHL   #1
                        57  DD 00325         PUSHL   R7
                68      03  FB 00327         CALLS   #3, DBG$PRINT
                        7E  D4 0032A 43$:    CLRL    -(SP)                                   ;0979
                        56  DD 0032C         PUSHL   R6
                        03  DD 0032E         PUSHL   #3
                        51  11 00330 44$:    BRB     49$
                        7E  D4 00332 45$:    CLRL    -(SP)                                   ;0985
                        56  DD 00334         PUSHL   R6
                        02  DD 00336         PUSHL   #2
                    14  AE  9F 00338         PUSHAB  POINTER
                6A      04  FB 0033B         CALLS   #4, FETCH_OPERAND
                0A      56  E9 0033E         BLBC    R6, 46$                                 ;0986
                    04  AE  9F 00341         PUSHAB  DELIMITER
                        01  DD 00344         PUSHL   #1
                        57  DD 00346         PUSHL   R7
                68      03  FB 00348         CALLS   #3, DBG$PRINT
                        7E  D4 0034B 46$:    CLRL    -(SP)                                   ;0987
                        56  DD 0034D         PUSHL   R6
                        03  DD 0034F         PUSHL   #3
                    14  AE  9F 00351         PUSHAB  POINTER
                6A      04  FB 00354         CALLS   #4, FETCH_OPERAND
                0A      56  E9 00357         BLBC    R6, 47$                                 ;0988
                    04  AE  9F 0035A         PUSHAB  DELIMITER
                        01  DD 0035D         PUSHL   #1
                        57  DD 0035F         PUSHL   R7
                68      03  FB 00361         CALLS   #3, DBG$PRINT
                        7E  D4 00364 47$:    CLRL    -(SP)                                   ;0989
                        56  DD 00366         PUSHL   R6
                        02  DD 00368         PUSHL   #2
                    14  AE  9F 0036A         PUSHAB  POINTER
                6A      04  FB 0036D         CALLS   #4, FETCH_OPERAND
                0A      56  E9 00370         BLBC    R6, 48$                                 ;0990
                    04  AE  9F 00373         PUSHAB  DELIMITER
                        01  DD 00376         PUSHL   #1
                        57  DD 00378         PUSHL   R7
                68      03  FB 0037A         CALLS   #3, DBG$PRINT
                        7E  D4 0037D 48$:    CLRL    -(SP)                                   ;0991
                        56  DD 0037F         PUSHL   R6
                        02  DD 00381         PUSHL   #2
```

```
                                013D  31  00383 49$:    BRW     70$
                          53          05  D0  00386 50$:    MOVL    #5, COUNT                                   ;0996
                                      7E  D4  00389 51$:    CLRL    -(SP)                                       ;0998
                                      56  DD  0038B         PUSHL   R6
                                      02  DD  0038D         PUSHL   #2
                                  14  AE  9F  0038F         PUSHAB  POINTER
                          6A          04  FB  00392         CALLS   #4, FETCH_OPERAND
                          0E          56  E9  00395         BLBC    R6, 52$                                     ;0999
                                      53  D5  00398         TSTL    COUNT
                                      0A  13  0039A         BEQL    52$
                                  04  AE  9F  0039C         PUSHAB  DELIMITER
                                      01  DD  0039F         PUSHL   #1
                                      57  DD  003A1         PUSHL   R7
                          68          03  FB  003A3         CALLS   #3, DBG$PRINT
                          E0          53  F4  003A6 52$:    SOBGEQ  COUNT, 51$                                  ;0996
                                      71  11  003A9         BRB     60$                                         ;0995
                                      7E  D4  003AB 53$:    CLRL    -(SP)                                       ;1008
                                      56  DD  003AD         PUSHL   R6
        53      09  A4                  00  EF  003AF         EXTZV   #0, #4, 9(OPCODE_ENTRY), R3
                                      53  DD  003B5         PUSHL   R3
                                  14  AE  9F  003B7         PUSHAB  POINTER
                          6A          04  FB  003BA         CALLS   #4, FETCH_OPERAND
                          0A          56  E9  003BD         BLBC    R6, 54$                                     ;1009
                                  04  AE  9F  003C0         PUSHAB  DELIMITER
                                      01  DD  003C3         PUSHL   #1
                                      57  DD  003C5         PUSHL   R7
                          68          03  FB  003C7         CALLS   #3, DBG$PRINT
                                      7E  D4  003CA 54$:    CLRL    -(SP)                                       ;1010
                                0048  8F  BB  003CC         PUSHR   #^M<R3,R6>
                                  14  AE  9F  003D0         PUSHAB  POINTER
                          6A          04  FB  003D3         CALLS   #4, FETCH_OPERAND
                          0A          56  E9  003D6         BLBC    R6, 55$                                     ;1011
                                  04  AE  9F  003D9         PUSHAB  DELIMITER
                                      01  DD  003DC         PUSHL   #1
                                      57  DD  003DE         PUSHL   R7
                          68          03  FB  003E0         CALLS   #3, DBG$PRINT
                                      7E  D4  003E3 55$:    CLRL    -(SP)                                       ;1016
                                0048  8F  BB  003E5         PUSHR   #^M<R3,R6>
                                  14  AE  9F  003E9         PUSHAB  POINTER
                          6A          04  FB  003EC         CALLS   #4, FETCH_OPERAND
                                      53  D5  003EF         TSTL    R3                                          ;1023
                                      05  12  003F1         BNEQ    56$
                          50          69  9A  003F3         MOVZBL  OP_BUFFER, LIMIT
                                      17  11  003F6         BRB     59$
                          01          53  D1  003F8 56$:    CMPL    R3, #1                                      ;1024
                                      05  12  003FB         BNEQ    57$
                          50          69  3C  003FD         MOVZWL  OP_BUFFER, LIMIT
                                      0D  11  00400         BRB     59$
                          02          53  D1  00402 57$:    CMPL    R3, #2                                      ;1025
                                      05  13  00405         BEQL    58$
                          50          01  CE  00407         MNEGL   #1, LIMIT
                                      03  11  0040A         BRB     59$
                          50          69  D0  0040C 58$:    MOVL    OP_BUFFER, LIMIT
                          0C          56  E8  0040F 59$:    BLBS    R6, 61$                                     ;1030
    08  AE          08  BE40  3E  00412         MOVAW   @POINTER[LIMIT], POINTER
    08  AE                  02  C0  00418         ADDL2   #2, POINTER
                            42  11  0041C 60$:    BRB     65$
```

```
                                55        08    AE   D0 0041E 61$:    MOVL    POINTER, START                                  ; 1034
                                53        01    A0   9E 00422          MOVAB   1(R0), COUNT                                    ; 1043
                                                35   11 00426          BRB     64$
                     09 00000000G  00             01   E1 00428 62$:    BBC     #1, DBG$GV_CONTROL+1, 63$                       ; 1038
                                        000280E8  8F   DD 00430          PUSHL   #164072
                                6B                01   FB 00436          CALLS   #1, LIB$SIGNAL
                                                  01   DD 00439 63$:    PUSHL   #1                                              ; 1040
                                          0C      AE   9F 0043B          PUSHAB  POINTER
                           0000V  CF              02   FB 0043E          CALLS   #2, FETCH_INSTRUCTION
                       00000000G  00              00   FB 00443          CALLS   #0, DBG$NEWLINE                                 ; 1041
                                          18      A7   9F 0044A          PUSHAB  P.AAH                                           ; 1042
                                                  02   DD 0044D          PUSHL   #2
                                                  57   DD 0044F          PUSHL   R7
                                68                03   FB 00451          CALLS   #3, DBG$PRINT
                                          00 B945 9F   9F 00454          PUSHAB  @CASE_OFFSET[START]                             ; 1043
                           0000V  CF              01   FB 00458          CALLS   #1, PRINT_ADDRESS
                                C8                53   F4 0045D 64$:    SOBGEQ  COUNT, 62$                                      ; 1035
                                                  67   11 00460 65$:    BRB     71$                                             ; 1005
                                                  7E   D4 00462 66$:    CLRL    -(SP)                                           ; 1051
                                                  56   DD 00464          PUSHL   R6
          7E        09    A4                    04   EF 00466          EXTZV   #4, #4, 9(OPCODE_ENTRY), -(SP)
                                          14      AE   9F 0046C          PUSHAB  POINTER
                                6A                04   FB 0046F          CALLS   #4, FETCH_OPERAND
                                0A                56   E9 00472          BLBC    R6, 67$                                         ; 1052
                                          04      AE   9F 00475          PUSHAB  DELIMITER
                                                  01   DD 00478          PUSHL   #1
                                                  57   DD 0047A          PUSHL   R7
                                68                03   FB 0047C          CALLS   #3, DBG$PRINT
                                                  7E   D4 0047F 67$:    CLRL    -(SP)                                           ; 1053
                                                  56   DD 00481          PUSHL   R6
          7E        09    A4                    04   EF 00483          EXTZV   #0, #4, 9(OPCODE_ENTRY), -(SP)
                                          14      AE   9F 00489          PUSHAB  POINTER
                                6A                04   FB 0048C          CALLS   #4, FETCH_OPERAND
                                0A                56   E9 0048F          BLBC    R6, 68$                                         ; 1054
                                          04      AE   9F 00492          PUSHAB  DELIMITER
                                                  01   DD 00495          PUSHL   #1
                                                  57   DD 00497          PUSHL   R7
                                68                03   FB 00499          CALLS   #3, DBG$PRINT
                                                  7E   D4 0049C 68$:    CLRL    -(SP)                                           ; 1055
                                                  56   DD 0049E          PUSHL   R6
          7E        09    A4                    04   EF 004A0          EXTZV   #4, #4, 9(OPCODE_ENTRY), -(SP)
                                          14      AE   9F 004A6          PUSHAB  POINTER
                                6A                04   FB 004A9          CALLS   #4, FETCH_OPERAND
                                0A                56   E9 004AC          BLBC    R6, 69$                                         ; 1056
                                          04      AE   9F 004AF          PUSHAB  DELIMITER
                                                  01   DD 004B2          PUSHL   #1
                                                  57   DD 004B4          PUSHL   R7
                                68                03   FB 004B6          CALLS   #3, DBG$PRINT
                                                  7E   D4 004B9 69$:    CLRL    -(SP)                                           ; 1057
                                                  56   DD 004BB          PUSHL   R6
          7E        09    A4                    00   EF 004BD          EXTZV   #0, #4, 9(OPCODE_ENTRY), -(SP)
                                          14      AE   9F 004C3 70$:    PUSHAB  POINTER
                                6A                04   FB 004C6          CALLS   #4, FETCH_OPERAND
                                50        08      AE   D0 004C9 71$:    MOVL    POINTER, R0                                     ; 1067
                                                  04   004CD            RET                                                     ; 1068
                                            0000  004CE 72$:    .WORD   Save nothing                                            ; 0786
                                50        08      AC   D0 004D0          MOVL    8(AP), R0                                       ;
```

```
                        50      04  A0  DD 004D4          MOVL    4(R0), R0
                                F8  A0  9F 004D8          PUSHAB  ERROR_VALUE
                                FC  A0  9F 004DB          PUSHAB  SIGNAL_FLAG
                                    02  DD 004DE          PUSHL   #2
                                    5E  DD 004E0          PUSHL   SP
                        7E      04  AC  7D 004E2          MOVQ    4(AP), -(SP)
                FAE1    CF          03  FB 004E6          CALLS   #3, DECODE_HANDLER
                                        04 004EB          RET

; Routine Size:  1260 bytes,    Routine Base:  DBG$CODE + 0034
```

```
;  953          1069  1 GLOBAL ROUTINE DBG$Ins_Encode(Input_Buffer,Output_Buffer,Relocation) =
;  954          1070  2     BEGIN
;  955          1071  2     MAP Input_Buffer    : REF VECTOR [,BYTE],   ! %ASCIC String
;  956          1072  2         Output_Buffer   : REF VECTOR [,BYTE];   ! Encoded instruction
;  957          1073  2     LOCAL
;  958          1074  2         Operand_number  : INITIAL( 0 ),                                     ! A007
;  959          1075  2         Encode          : BLOCK [20,BYTE] FIELD(Encode_Fields),
;  960          1076  2         Local_Buffer    : VECTOR[256,BYTE],
;  961          1077  2         Opcode_Entry    : REF BLOCK[10,BYTE] FIELD(Opcode_Entry_Fields),
;  962          1078  2         State;
;  963          1079  2
;  964          1080  2     Encode[Enc_Input_Class]   =  dsc$k_class_s;
;  965          1081  2     Encode[Enc_Input_Dtype]   =  dsc$k_dtype_t;
;  966          1082  2     Encode[Enc_Input_Length]  = .Input_Buffer[0];
;  967          1083  2     Encode[Enc_Input_Buffer]  =  Local_Buffer[0];
;  968          1084  2     Encode[Enc_Output_Length] =  0;
;  969          1085  2     Encode[Enc_Output_Buffer] =  Output_Buffer[1];
;  970          1086  2     Encode[Enc_Final_Address] = .Relocation;
;  971          1087  2
;  972          1088  2     ch$move(.Input_Buffer[0],Input_Buffer[1],Local_Buffer[0]);
;  973          1089  2     Opcode_Entry = Opcode_Name_Index(Encode[Enc_Input_Desc],%C' ');          ! Changed to call Opcode_Name
;  974          1090  2                                                                             ! instead of DBG$OPCODE_INDEX
;  975          1091  2     Opcode_Entry = DBG$Opcode_Name_Table[.Opcode_Entry,offset];
;  976          1092  2
;  977          1093  2     IF .Opcode_Entry[op_code_one] NEQ 0 THEN
;  978          1094  2     Store_Operand(Encode,Opcode_Entry[op_code_one],1);
;  979          1095  2     Store_Operand(Encode,Opcode_Entry[op_code_two],1);
;  980          1096  2
;  981          1097  2     State = .Opcode_Entry[op_kind];
;  982          1098  2
;  983          1099  2     WHILE (.State NEQ simple_0_operand) DO
;  984          1100  3         BEGIN
;  985          1101  3         CASE .State FROM simple_1_operand TO maximum_state OF
;  986          1102  3             SET
;  987          1103  3
;  988          1104  3             [simple_1_operand,
;  989          1105  3              simple_2_operand,
;  990          1106  3              simple_3_operand]:
;  991          1107  4                 BEGIN
;  992          1108  4                 Parse_Operand(Encode,.Opcode_Entry[op_type_one],Operand_number);! M007
;  993          1109  4                 State = .State - 1;
;  994          1110  3                 END;
;  995          1111  3
;  996          1112  3             [branch_0_operand]:
;  997          1113  4                 BEGIN
;  998          1114  4                 LOCAL Address,Length;
;  999          1115  4                 Operand_number = .Operand_number + 1;                   ! A007
; 1000          1116  4                 Length = Scan_Operand(Encode[Enc_Input_Desc],%C',');
; 1001          1117  4                 IF NOT Parse_Expression(-1,.Encode[Enc_Input_Buffer],
; 1002          1118  4                  .Encode[Enc_Input_Length],Address) THEN SIGNAL(dbg$_INVEXPR,1,.Operand_number);     ! M0
; 1003          1119  4                 IF (Check_Register(.Address) GEQ 0) THEN SIGNAL(dbg$_INVEXPR,1,.Operand_number);    ! M0
; 1004          1120  4                 Encode[Enc_Input_Buffer] = .Encode[Enc_Input_Buffer] + .Length;
; 1005          1121  4                 Encode[Enc_Input_Length] = .Encode[Enc_Input_Length] - .Length;
; 1006          1122  4
; 1007          1123  4                 Length = .Data_Size[.Opcode_Entry[op_type_one]];
; 1008          1124  4
; 1009          1125  5                 Address = .Address - (.Length + .Encode[Enc_Final_Address]
```

```
;1010     1126  4                                   + .Encode[Enc_Output_Length]);
;1011     1127  4                 Store_Operand(Encode,Address,.Length);
;1012     1128  4                 EXITLOOP;
;1013     1129  4                 END;
;1014     1130  4
;1015     1131  4               [branch_1_operand,
;1016     1132  4                branch_2_operand,
;1017     1133  4                branch_3_operand]:
;1018     1134  4                 BEGIN
;1019     1135  4                 Parse_Operand(Encode,.Opcode_Entry[op_type_two],Operand_number);! M007
;1020     1136  4                 State = .State - 1;
;1021     1137  4                 END;
;1022     1138  4
;1023     1139  4               [convert_datatype,
;1024     1140  4  ! *****       evaluate_address,
;1025     1141  4  ! *****       simple_bit_field,
;1026     1142  4  ! *****       routine_dispatch,
;1027     1143  4  ! *****       polynomial_value,
;1028     1144  4  ! *****       probe_for_access,
;1029     1145  4                string_3_operand]:
;1030     1146  4                 BEGIN
;1031     1147  4                 Parse_Operand(Encode,.Opcode_Entry[op_type_two],Operand_number);! M007
;1032     1148  4                 State = Simple_1_Operand;
;1033     1149  4                 END;
;1034     1150  4
;1035     1151  4               [string_4_operand]:
;1036     1152  4                 BEGIN
;1037     1153  4                 Parse_Operand(Encode,.Opcode_Entry[op_type_two],Operand_number);! M007
;1038     1154  4                 State = Simple_2_Operand;
;1039     1155  4                 END;
;1040     1156  4
;1041     1157  4               [string_5_operand,
;1042     1158  4                string_6_operand]:
;1043     1159  4                 BEGIN
;1044     1160  4                 Parse_Operand(Encode,.Opcode_Entry[op_type_two],Operand_number);! M007
;1045     1161  4                 DECR count FROM .State TO string_5_operand DO
;1046     1162  4                     Parse_Operand(Encode,context_b,Operand_number);             ! M007
;1047     1163  4                 State = Simple_1_Operand;
;1048     1164  3                 END;
;1049     1165  3
;1050     1166  3               [trailing_operand]:
;1051     1167  4                 BEGIN
;1052     1168  4  !              Store_Operand(Encode,.Opcode_Entry[op_type_one]);
;1053     1169  4  !              IF .Print_Flag THEN Print_Operand(.Opcode_Entry[op_type_one]);
;1054     1170  4                 EXITLOOP;
;1055     1171  3                 END;
;1056     1172  3
;1057     1173  3               [complex_SHIFT]:
;1058     1174  4                 BEGIN
;1059     1175  4                 Parse_Operand(Encode,context_b,Operand_number);         ! M007
;1060     1176  4                 State = Simple_2_Operand;
;1061     1177  3                 END;
;1062     1178  3
;1063     1179  3               [complex_EMOD]:
;1064     1180  4                 BEGIN
;1065     1181  4                 Parse_Operand(Encode,.Opcode_Entry[op_type_one],Operand_number);! M007
;1066     1182  4                 Parse_Operand(Encode,.Opcode_Entry[op_type_two],Operand_number);! M007
```

```
;  1067          1183  4              State = Simple_3_Operand;
;  1068          1184  3              END;
;  1069          1185  3
;  1070          1186  3          [complex_CRC]:
;  1071          1187  4              BEGIN
;  1072          1188  4              Parse_Operand(Encode,context_b,Operand_number);          ! M007
;  1073          1189  4              Parse_Operand(Encode,context_l,Operand_number);          ! M007
;  1074          1190  4              Parse_Operand(Encode,context_t,Operand_number);          ! M007
;  1075          1191  4              EXITLOOP;
;  1076          1192  3              END;
;  1077          1193  3
;  1078          1194  3          [complex_EMUL]:
;  1079          1195  4              BEGIN
;  1080          1196  4              Parse_Operand(Encode,context_l,Operand_number);          ! M007
;  1081          1197  4              Parse_Operand(Encode,context_l,Operand_number);          ! M007
;  1082          1198  4              Parse_Operand(Encode,context_l,Operand_number);          ! M007
;  1083          1199  4              Parse_Operand(Encode,context_q,Operand_number);          ! M007
;  1084          1200  4              EXITLOOP;
;  1085          1201  3              END;
;  1086          1202  3
;  1087          1203  3          [complex_EDIV]:
;  1088          1204  4              BEGIN
;  1089          1205  4              Parse_Operand(Encode,context_l,Operand_number);          ! M007
;  1090          1206  4              Parse_Operand(Encode,context_q,Operand_number);          ! M007
;  1091          1207  4              Parse_Operand(Encode,context_l,Operand_number);          ! M007
;  1092          1208  4              Parse_Operand(Encode,context_l,Operand_number);          ! M007
;  1093          1209  4              EXITLOOP;
;  1094          1210  3              END;
;  1095          1211  3
;  1096          1212  3          [complex_INDEX]:
;  1097          1213  4              BEGIN
;  1098          1214  4              DECR count FROM 5 TO 0 DO Parse_Operand(Encode,context_l,Operand_number);! M007
;  1099          1215  4              EXITLOOP;
;  1100          1216  3              END;
;  1101          1217  3
;  1102          1218  3          [complex_CASE]:
;  1103          1219  4              BEGIN
;  1104          1220  4              Parse_Operand(Encode,.Opcode_Entry[op_type_one],Operand_number);! M007
;  1105          1221  4              Parse_Operand(Encode,.Opcode_Entry[op_type_one],Operand_number);! M007
;  1106          1222  4              Parse_Operand(Encode,.Opcode_Entry[op_type_one],Operand_number);! M007
;  1107          1223  4              EXITLOOP;
;  1108          1224  3              END;
;  1109          1225  3
;  1110          1226  3          [complex_ASHP]:                                                 !
;  1111          1227  4              BEGIN                                                       !
;  1112          1228  4              Parse_Operand(Encode,.Opcode_Entry[op_type_two],Operand_number);!
;  1113          1229  4              Parse_Operand(Encode,.Opcode_Entry[op_type_one],Operand_number);                 !
;  1114          1230  4              Parse_Operand(Encode,.Opcode_Entry[op_type_two],Operand_number);                 !
;  1115          1231  4              Parse_Operand(Encode,.Opcode_Entry[op_type_one],Operand_number);                 !
;  1116          1232  4              EXITLOOP;                                                   !
;  1117          1233  3              END;                                                        !
;  1118          1234  3
;  1119          1235  3          [INRANGE,OUTRANGE]:
;  1120          1236  3              $DBG_ERROR('DBG$Encode_Decode - bad opcode table entry');
;  1121          1237  3
;  1122          1238  3          TES;
;  1123          1239  2      END;
```

```
; 1124          1240  2
; 1125          1241  2        Output_Buffer[0] = .Encode[Enc_Output_Length];
; 1126          1242  2        RETURN 1;
; 1127          1243  1        END;


                                                                     .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

63 65 44 5F 65 64 6F 63 6E 45 24 47 42 44 2A 0114D P.AAJ:  .ASCII  \*DBG$Encode_Decode - bad opcode table en\  ;
64 6F 63 70 6F 20 64 61 62 20 2D 20 65 64 6F 0115C                                                         ;
               6E 65 20 65 6C 62 61 74 20 65 0116B                                                         ;
                                  79 72 74 01175           .ASCII  \try\                                    ;


                                                                     .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                                     03FC 00000            .ENTRY  DBG$INS_ENCODE, Save R2,R3,R4,R5,R6,R7,R8,- ; 1069
                                                                   R9                                          ;
                  59     0000V   CF   9E 00002             MOVAB   STORE_OPERAND, R9                           ;
                  58 00000000G   00   9E 00007             MOVAB   LIB$SIGNAL, R8                              ;
                  57 00000000'   EF   9E 0000E             MOVAB   DBG$OPCODE_NAME_TABLE, R7                   ;
                  56     0000V   CF   9E 00015             MOVAB   PARSE_OPERAND, R6                           ;
                  5E      FEE4   CE   9E 0001A             MOVAB   -284(SP), SP                                ;
                            04   AE   D4 0001F             CLRL    OPERAND_NUMBER                              ; 1070
             EE   AD      010E   8F   B0 00022             MOVW    #270, ENCODE+2                              ; 1081
                  50        04   AC   D0 00028             MOVL    INPUT_BUFFER, R0                            ; 1082
             EC   AD             60   9B 0002C             MOVZBW  (R0), ENCODE                                ;
             F0   AD        08   AE   9E 00030             MOVAB   LOCAL_BUFFER, ENCODE+4                      ; 1083
                            F4   AD   D4 00035             CLRL    ENCODE+8                                    ; 1084
   F8   AD   08   AC             01   C1 00038             ADDL3   #1, OUTPUT_BUFFER, ENCODE+12                ; 1085
             FC   AD        0C   AC   D0 0003E             MOVL    RELOCATION, ENCODE+16                       ; 1086
                  51             60   9A 00043             MOVZBL  (R0), R1                                    ; 1088
   08   AE   01   A0             51   28 00046             MOVC3   R1, 1(R0), LOCAL_BUFFER                     ;
                                 20   DD 0004C             PUSHL   #32                                         ; 1089
                            EC   AD   9F 0004E             PUSHAB  ENCODE                                      ;
           0000V  CF             02   FB 00051             CALLS   #2, OPCODE_NAME_INDEX                       ;
                  54             50   D0 00056             MOVL    R0, OPCODE_ENTRY                            ;
        50        54             0A   C5 00059             MULL3   #10, OPCODE_ENTRY, R0                       ; 1091
        54        50             57   C1 0005D             ADDL3   R7, R0, OPCODE_ENTRY                        ;
                            06   A4   95 00061             TSTB    6(OPCODE_ENTRY)                             ; 1093
                                 0B   13 00064             BEQL    1$                                          ;
                                 01   DD 00066             PUSHL   #1                                          ; 1094
                            06   A4   9F 00068             PUSHAB  6(OPCODE_ENTRY)                             ;
                            EC   AD   9F 0006B             PUSHAB  ENCODE                                      ;
                  69             03   FB 0006E             CALLS   #3, STORE_OPERAND                           ;
                                 01   DD 00071 1$:         PUSHL   #1                                          ; 1095
                            07   A4   9F 00073             PUSHAB  7(OPCODE_ENTRY)                             ;
                            EC   AD   9F 00076             PUSHAB  ENCODE                                      ;
                  69             03   FB 00079             CALLS   #3, STORE_OPERAND                           ;
                  53        08   A4   9A 0007C             MOVZBL  8(OPCODE_ENTRY), STATE                      ; 1097
                                 53   D5 00080 2$:         TSTL    STATE                                       ; 1099
                                 03   12 00082             BNEQ    3$                                          ;
                               022C   31 00084             BRW     32$                                         ;
        16        01             53   CF 00087 3$:         CASEL   STATE, #1, #22                              ; 1101
   004B      003F      003F    003F      0008B 4$:         .WORD   6$-4$,-                                     ;
```

```
                00D7        00C4        00C4        00C4    00093                   6$-4$,-
                00F3        00E8        00D7        0228    0009B                   6$-4$,-
                011E        002E        002E        00F3    000A3                   7$-4$,-
                016F        012E        0197        01D2    000AB                   10$-4$,-
                            01EC        0152        01BF    000B3                   10$-4$,-
                                                                                    10$-4$,-
                                                                                    12$-4$,-
                                                                                    32$-4$,-
                                                                                    12$-4$,-
                                                                                    13$-4$,-
                                                                                    14$-4$,-
                                                                                    14$-4$,-
                                                                                    5$-4$,-
                                                                                    5$-4$,-
                                                                                    18$-4$,-
                                                                                    28$-4$,-
                                                                                    24$-4$,-
                                                                                    20$-4$,-
                                                                                    23$-4$,-
                                                                                    26$-4$,-
                                                                                    22$-4$,-
                                                                                    29$-4$
                                    114D    C7  9F 000B9 5$:    PUSHAB  P.AAJ                   1236
                                            01  DD 000BD         PUSHL   #1
                                00028362    8F  DD 000BF         PUSHL   #164706
                                        68      03  FB 000C5         CALLS   #3, LIB$SIGNAL
                                            B6  11 000C8         BRB     2$
                                        04  AE  9F 000CA 6$:     PUSHAB  OPERAND_NUMBER          1108
                    7E      09  A4          04      00  EF 000CD         EXTZV   #0, #4, 9(OPCODE_ENTRY), -(SP)
                                            0082    31  000D3         BRW     11$
                                        04  AE  D6 000D6 7$:     INCL    OPERAND_NUMBER          1115
                                            2C  DD 000D9         PUSHL   #44                     1116
                                        EC  AD  9F 000DB         PUSHAB  ENCODE
                                0000V   CF          02  FB 000DE         CALLS   #2, SCAN_OPERAND
                                        52          50  D0 000E3         MOVL    R0, LENGTH
                                                    5E  DD 000E6         PUSHL   SP              1117
                                        7E      EC  AD  3C 000E8         MOVZWL  ENCODE, -(SP)   1118
                                            F0  AD  DD 000EC         PUSHL   ENCODE+4        1117
                                        7E          01  CE 000EF         MNEGL   #1, -(SP)
                                0000V   CF          04  FB 000F2         CALLS   #4, PARSE_EXPRESSION
                                        0E          50  E8 000F7         BLBS    R0, 8$
                                        04  AE      DD 000FA         PUSHL   OPERAND_NUMBER  1118
                                            01      DD 000FD         PUSHL   #1
                                00028290    8F      DD 000FF         PUSHL   #164496
                                        68          03  FB 00105         CALLS   #3, LIB$SIGNAL
                                                    6E  DD 00108 8$:     PUSHL   ADDRESS         1119
                                0000V   CF          01  FB 0010A         CALLS   #1, CHECK_REGISTER
                                                    50  D5 0010F         TSTL    R0
                                                    0E  19 00111         BLSS    9$
                                        04  AE      DD 00113         PUSHL   OPERAND_NUMBER
                                            01      DD 00116         PUSHL   #1
                                00028290    8F      DD 00118         PUSHL   #164496
                                        68          03  FB 0011E         CALLS   #3, LIB$SIGNAL
                                    F0  AD          52  C0 00121 [illegible]  ADDL2   LENGTH, ENCODE+4  1120
                                    EC  AD          52  A2 00125 [illegible]  SUBW2   LENGTH, ENCODE    1121
                    50      09  A4          04      00  EF 0012[illegible]  EXTZV   #0, #4, 9(OPCODE_ENTRY), R0  1123
                                            52  10A0 C740   9A 0012F     MOVZBL  DATA_SIZE[R0], LENGTH
```

```
                        50          52      FC    AD   C1 00135          ADDL3   ENCODE+16, LENGTH, R0                    ;1125
                                    50      F4    AD   C0 0013A          ADDL2   ENCODE+8, R0                             ;1126
                                    6E            50   C2 0013E          SUBL2   R0, ADDRESS                              ;1127
                                                  52   DD 00141          PUSHL   LENGTH
                                          04      AE   9F 00143          PUSHAB  ADDRESS
                                          EC      AD   9F 00146          PUSHAB  ENCODE
                                                  69   03 FB 00149          CALLS   #3, STORE_OPERAND
                                                0164   31 0014C          BRW     32$                                      ;1113
                                          04      AE   9F 0014F 10$:     PUSHAB  OPERAND_NUMBER                           ;1135
                7E      09      A4                04   04 EF 00152          EXTZV   #4, #4, 9(OPCODE_ENTRY), -(SP)
                                          EC      AD   9F 00158 11$:     PUSHAB  ENCODE
                                                  66   03 FB 0015B          CALLS   #3, PARSE_OPERAND
                                                       53 D7 0015E          DECL    STATE                                    ;1136
                                                       78 11 00160          BRB     21$                                      ;1101
                                          04      AE   9F 00162 12$:     PUSHAB  OPERAND_NUMBER                           ;1147
                7E      09      A4                04   04 EF 00165          EXTZV   #4, #4, 9(OPCODE_ENTRY), -(SP)
                                          EC      AD   9F 0016B          PUSHAB  ENCODE
                                                  66   03 FB 0016E          CALLS   #3, PARSE_OPERAND
                                                       31 11 00171          BRB     17$                                      ;1148
                                          04      AE   9F 00173 13$:     PUSHAB  OPERAND_NUMBER                           ;1153
                7E      09      A4                04   04 EF 00176          EXTZV   #4, #4, 9(OPCODE_ENTRY), -(SP)
                                                       30 11 0017C          BRB     19$
                                          04      AE   9F 0017E 14$:     PUSHAB  OPERAND_NUMBER                           ;1160
                7E      09      A4                04   04 EF 00181          EXTZV   #4, #4, 9(OPCODE_ENTRY), -(SP)
                                          EC      AD   9F 00187          PUSHAB  ENCODE
                                                  66   03 FB 0018A          CALLS   #3, PARSE_OPERAND
                                                  52   53 D0 0018D          MOVL    STATE, COUNT                             ;1161
                                                       0D 11 00190          BRB     16$
                                          04      AE   9F 00192 15$:     PUSHAB  OPERAND_NUMBER                           ;1162
                                                       7E D4 00195          CLRL    -(SP)
                                          EC      AD   9F 00197          PUSHAB  ENCODE
                                                  66   03 FB 0019A          CALLS   #3, PARSE_OPERAND
                                                       52 D7 0019D          DECL    COUNT
                                                  0C   52 D1 0019F 16$:     CMPL    COUNT, #12
                                                       EE 18 001A2          BGEQ    15$
                                                  53   01 D0 001A4 17$:     MOVL    #1, STATE                                ;1163
                                                       31 11 001A7          BRB     21$                                      ;1101
                                          04      AE   9F 001A9 18$:     PUSHAB  OPERAND_NUMBER                           ;1175
                                                       7E D4 001AC          CLRL    -(SP)
                                          EC      AD   9F 001AE 19$:     PUSHAB  ENCODE
                                                  66   03 FB 001B1          CALLS   #3, PARSE_OPERAND
                                                  53   02 D0 001B4          MOVL    #2, STATE                                ;1176
                                                       21 11 001B7          BRB     21$                                      ;1101
                                          04      AE   9F 001B9 20$:     PUSHAB  OPERAND_NUMBER                           ;1181
                7E      09      A4                04   00 EF 001BC          EXTZV   #0, #4, 9(OPCODE_ENTRY), -(SP)
                                          EC      AD   9F 001C2          PUSHAB  ENCODE
                                                  66   03 FB 001C5          CALLS   #3, PARSE_OPERAND
                                          04      AE   9F 001C8          PUSHAB  OPERAND_NUMBER                           ;1182
                7E      09      A4                04   04 EF 001CB          EXTZV   #4, #4, 9(OPCODE_ENTRY), -(SP)
                                          EC      AD   9F 001D1          PUSHAB  ENCODE
                                                  66   03 FB 001D4          CALLS   #3, PARSE_OPERAND
                                                  53   03 D0 001D7          MOVL    #3, STATE                                ;1183
                                                FEA3   31 001DA 21$:     BRW     2$                                       ;1101
                                          04      AE   9F 001DD 22$:     PUSHAB  OPERAND_NUMBER                           ;1188
                                                       7E D4 001E0          CLRL    -(SP)
                                          EC      AD   9F 001E2          PUSHAB  ENCODE
                                                  66   03 FB 001E5          CALLS   #3, PARSE_OPERAND
```

```
                        04 AE 9F 001E8          PUSHAB  OPERAND_NUMBER                          ; 1189
                        02 DD 001EB             PUSHL   #2
                     EC AD 9F 001ED             PUSHAB  ENCODE
                  66    03 FB 001F0             CALLS   #3, PARSE_OPERAND
                        04 AE 9F 001F3          PUSHAB  OPERAND_NUMBER                          ; 1190
                        0B DD 001F6             PUSHL   #11
                        4E 11 001F8             BRB     25$
                        04 AE 9F 001FA 23$:     PUSHAB  OPERAND_NUMBER                          ; 1196
                        02 DD 001FD             PUSHL   #2
                     EC AD 9F 001FF             PUSHAB  ENCODE
                  66    03 FB 00202             CALLS   #3, PARSE_OPERAND
                        04 AE 9F 00205          PUSHAB  OPERAND_NUMBER                          ; 1197
                        02 DD 00208             PUSHL   #2
                     EC AD 9F 0020A             PUSHAB  ENCODE
                  66    03 FB 0020D             CALLS   #3, PARSE_OPERAND
                        04 AE 9F 00210          PUSHAB  OPERAND_NUMBER                          ; 1198
                        02 DD 00213             PUSHL   #2
                     EC AD 9F 00215             PUSHAB  ENCODE
                  66    03 FB 00218             CALLS   #3, PARSE_OPERAND
                        04 AE 9F 0021B          PUSHAB  OPERAND_NUMBER                          ; 1199
                        03 DD 0021E             PUSHL   #3
                        26 11 00220             BRB     25$
                        04 AE 9F 00222 24$:     PUSHAB  OPERAND_NUMBER                          ; 1205
                        02 DD 00225             PUSHL   #2
                     EC AD 9F 00227             PUSHAB  ENCODE
                  66    03 FB 0022A             CALLS   #3, PARSE_OPERAND
                        04 AE 9F 0022D          PUSHAB  OPERAND_NUMBER                          ; 1206
                        03 DD 00230             PUSHL   #3
                     EC AD 9F 00232             PUSHAB  ENCODE
                  66    03 FB 00235             CALLS   #3, PARSE_OPERAND
                        04 AE 9F 00238          PUSHAB  OPERAND_NUMBER                          ; 1207
                        02 DD 0023B             PUSHL   #2
                     EC AD 9F 0023D             PUSHAB  ENCODE
                  66    03 FB 00240             CALLS   #3, PARSE_OPERAND
                        04 AE 9F 00243          PUSHAB  OPERAND_NUMBER                          ; 1208
                        02 DD 00246             PUSHL   #2
                        63 11 00248 25$:        BRB     31$
                  52    05 D0 0024A 26$:        MOVL    #5, COUNT                               ; 1214
                        04 AE 9F 0024D 27$:     PUSHAB  OPERAND_NUMBER
                        02 DD 00250             PUSHL   #2
                     EC AD 9F 00252             PUSHAB  ENCODE
                  66    03 FB 00255             CALLS   #3, PARSE_OPERAND
                  F2    52 F4 00258             SOBGEQ  COUNT, 27$
                        56 11 0025B             BRB     32$                                     ; 1213
                        04 AE 9F 0025D 28$:     PUSHAB  OPERAND_NUMBER                          ; 1220
7E      09  A4      04    00 EF 00260             EXTZV   #0, #4, 9(OPCODE_ENTRY), -(SP)
                     EC AD 9F 00266             PUSHAB  ENCODE
                  66    03 FB 00269             CALLS   #3, PARSE_OPERAND
                        04 AE 9F 0026C          PUSHAB  OPERAND_NUMBER                          ; 1221
7E      09  A4      04    00 EF 0026F             EXTZV   #0, #4, 9(OPCODE_ENTRY), -(SP)
                        27 11 00275             BRB     30$
                        04 AE 9F 00277 29$:     PUSHAB  OPERAND_NUMBER                          ; 1228
7E      09  A4      04    04 EF 0027A             EXTZV   #4, #4, 9(OPCODE_ENTRY), -(SP)
                     EC AD 9F 00280             PUSHAB  ENCODE
                  66    03 FB 00283             CALLS   #3, PARSE_OPERAND
                        04 AE 9F 00286          PUSHAB  OPERAND_NUMBER                          ; 1229
7E      09  A4      04    00 EF 00289             EXTZV   #0, #4, 9(OPCODE_ENTRY), -(SP)
```

```
                                                EC  AD  9F 0028F          PUSHAB  ENCODE                                          ;
                                             66     03  FB 00292          CALLS   #3, PARSE_OPERAND                             ;
                                                04  AE  9F 00295          PUSHAB  OPERAND_NUMBER                                ; 1230
         7E      09   A4                    04     04  EF 00298          EXTZV   #4, #4, 9(OPCODE_ENTRY), -(SP)                ;
                                                EC  AD  9F 0029E 30$:     PUSHAB  ENCODE                                        ;
                                             66     03  FB 002A1          CALLS   #3, PARSE_OPERAND                             ;
                                                04  AE  9F 002A4          PUSHAB  OPERAND_NUMBER                                ; 1231
         7E      09   A4                    04     00  EF 002A7          EXTZV   #0, #4, 9(OPCODE_ENTRY), -(SP)                ;
                                                EC  AD  9F 002AD 31$:     PUSHAB  ENCODE                                        ;
                                             66     03  FB 002B0          CALLS   #3, PARSE_OPERAND                             ;
                                08   BC      F4  AD  90 002B3 32$:     MOVB    ENCODE+8, @OUTPUT_BUFFER                      ; 1241
                                50             01  D0 002B8          MOVL    #1, R0                                        ; 1242
                                                   04 002BB          RET                                                   ; 1243

; Routine Size:  700 bytes,    Routine Base:  DBG$CODE + 0520
```

```
;  1129          1244  1 GLOBAL ROUTINE DBG$Opcode_Index(Input_Desc : REF dbg$stg_desc) =
;  1130          1245  1 !++                                                                                    !
;  1131          1246  1 !                                                                                      !
;  1132          1247  1 !       Return the index of DBG$Opcode_Name_Table of the opcode MNemonic               !
;  1133          1248  1 !       Passed in Input_Desc.                                                          !
;  1134          1249  1 !                                                                                      !
;  1135          1250  1 !       Input_desc is modified to point after the MNemonic.  That is the               !
;  1136          1251  1 !       MNemonic is consumed.  Input_desc must be the Mnemonic only if                 !
;  1137          1252  1 !       Delimiter is not passed.                                                       !
;  1138          1253  1 !                                                                                      !
;  1139          1254  1 !--                                                                                    !
;  1140          1255  2     BEGIN                                                                              !
;  1141          1256  2     LOCAL                                                                              !
;  1142          1257  2         Code,                                                                          !
;  1143          1258  2         Index;                                                                         !
;  1144          1259  2                                                                                        !
;  1145          1260  2     Index = Opcode_Name_Index ( .Input_desc );                                         ! Get the index into the name
;  1146          1261  2     Code = .DBG$Opcode_Name_Table[ .index, 7, 0, 8, 0];                                !
;  1147          1262  2     !++                                                                                !
;  1148          1263  2     ! Return the index into the Kind_table                                             !
;  1149          1264  2     !--                                                                                !
;  1150          1265  4     RETURN (IF (.DBG$Opcode_Name_Table[ .index, 6, 0, 8, 0] LSS %X'FD')               ! Test for 2 byte opcode!
;  1151          1266  4             THEN (.Code)                                                               ! 1 byte opcode index
;  1152          1267  4             ELSE ( .Code + %X'100' )                                                   ! 2 byte opcode index
;  1153          1268  2           );                                                                           !
;  1154          1269  1     END;


                                                0000 00000          .ENTRY  DBG$OPCODE_INDEX, Save nothing          ; 1244
                                     04   AC    DD 00002          PUSHL   INPUT_DESC                              ; 1260
                          0000V      CF         01 FB 00005          CALLS   #1, OPCODE_NAME_INDEX                  ;
                                     50         0A C4 0000A          MULL2   #10, R0                             ; 1261
                                     51 00000000'EF40 9A 0000D          MOVZBL  DBG$OPCODE_NAME_TABLE+7[R0], CODE
                                FD   8F 00000000'EF40 91 00015          CMPB    DBG$OPCODE_NAME_TABLE+6[R0], #253   ; 1265
                                                04 1E 0001E          BGEQU   1$                                  ;
                                     50         51 D0 00020          MOVL    CODE, R0                            ; 1266
                                                04 00023          RET                                           ;
                                     50    0100 C1 9E 00024 1$:     MOVAB   256(R1), R0                         ; 1267
                                                04 00029          RET                                           ; 1269

; Routine Size:  42 bytes,    Routine Base:  DBG$CODE + 07DC
```

```
;  1156          1270  1 ROUTINE Opcode_Name_Index(Input_Desc : REF dbg$stg_desc,Delimiter) =
;  1157          1271  1 !
;  1158          1272  1 ! FUNCTIONAL DESCRIPTION:
;  1159          1273  1 !
;  1160          1274  1 !   Opcode_Name_Index returns the index into DBG$OPCODE_NAME_TABLE
;  1161          1275  1 !   given a mnemonic.
;  1162          1276  1 !
;  1163          1277  1 ! FORMAL PARAMETERS:
;  1164          1278  1 !
;  1165          1279  1 !   Input_Desc - A string descriptor by reference, describing the
;  1166          1280  1 !                Mnemonic.
;  1167          1281  1 !
;  1168          1282  1 !   Delimiter  - An optional parameter, describing the character
;  1169          1283  1 !                with which to terminate the Mnemonic, passed
;  1170          1284  1 !                by value.
;  1171          1285  1 !
;  1172          1286  1 ! IMPLICIT INPUTS:
;  1173          1287  1 !
;  1174          1288  1 !   DBG$Opcode_Name_table
;  1175          1289  1 !
;  1176          1290  1 ! IMPLICIT OUTPUTS:
;  1177          1291  1 !
;  1178          1292  1 !   Input_desc is update to consume the Mnemonic.
;  1179          1293  1 !
;  1180          1294  1 !
;  1181          1295  1 ! ROUTINE VALUE:
;  1182          1296  1 !
;  1183          1297  1 !   The index into DBG$Opcode_Name_Table for the Mnemonic passed
;  1184          1298  1 !
;  1185          1299  1 !
;  1186          1300  1 ! SIDE EFFECTS:
;  1187          1301  1 !
;  1188          1302  1 !   None
;  1189          1303  1 !
;  1190          1304  2     BEGIN
;  1191          1305  2     BUILTIN ACTUALCOUNT;
;  1192          1306  2     LOCAL
;  1193          1307  2         index,
;  1194          1308  2         length,
;  1195          1309  2         limit_LSS,
;  1196          1310  2         limit_GTR;
;  1197          1311  2
;  1198          1312  2
;  1199          1313  2     IF Actualcount() GTR 1
;  1200          1314  2     THEN
;  1201          1315  2         length = Scan_Operand(.Input_Desc,.Delimiter)
;  1202          1316  2     ELSE
;  1203          1317  2         BEGIN
;  1204          1318  2
;  1205          1319  2         LOCAL Temp_ptr;                                    ! Current position in the inp
;  1206          1320  2
;  1207          1321  2         Skip_Leading_Blanks(.Input_Desc);
;  1208          1322  2
;  1209          1323  2         Temp_ptr = CH$PTR ( .Input_Desc[ dsc$a_pointer ] ); ! Assign the start of the str
;  1210          1324  2
;  1211          1325  3         !++
;  1212          1326  3         ! Consume the Mnemonic
```

```
;  1213          1327  3          !--
;  1214          1328  3          length = 0;
;  1215          1329  3          WHILE .length LSS .Input_Desc[dsc$w_length] DO
;  1216          1330  4              BEGIN
;  1217          1331  4              LOCAL char : BYTE UNSIGNED;
;  1218          1332  4              char = CH$RCHAR_A( Temp_ptr );                          ! Read a character and incr t
;  1219          1333  5              IF ((.char GEQU %C'A') AND (.char LEQU %C'Z'))
;  1220          1334  5              OR ((.char GEQU %C'0') AND (.char LEQU %C'9'))
;  1221          1335  4              THEN
;  1222          1336  4                  length = .length + 1
;  1223          1337  4              ELSE
;  1224          1338  4                  EXITLOOP;
;  1225          1339  3              END;
;  1226          1340  2          END;
;  1227          1341  2
;  1228          1342  2      limit_LSS = 0;
;  1229          1343  2      limit_GTR = Opcode_Index + 1;
;  1230          1344  2      !++                                                             !
;  1231          1345  2      !   Binary search of DBG$Opcode_Name_Table for the Mnemonic     !
;  1232          1346  2      !--                                                             !
;  1233          1347  2      WHILE ((index = (.limit_LSS + .limit_GTR)/2) NEQ .limit_LSS) DO
;  1234          1348  3          BEGIN
;  1235          1349  3          SELECTONE ch$compare(6,DBG$Opcode_Name_Table[.index,offset],
;  1236          1350                            .length,.Input_Desc[dsc$a_pointer],%C' ') OF
;  1237          1351  3              SET
;  1238          1352  3              [-1]: limit_LSS = .index;
;  1239          1353  3              [+1]: limit_GTR = .index;
;  1240          1354  3              [ 0]:
;  1241          1355  4                  BEGIN
;  1242          1356  4                  Input_Desc[dsc$w_length]  = .Input_Desc[dsc$w_length] -.length;
;  1243          1357  4                  Input_Desc[dsc$a_pointer] = .Input_Desc[dsc$a_pointer]+.length;
;  1244          1358  4                  RETURN .Index
;  1245          1359  3                  END;
;  1246          1360  3              TES;
;  1247          1361  2          END;
;  1248          1362  2      SIGNAL(dbg$_badopCode,2,.length,.Input_Desc[dsc$a_pointer]);
;  1249          1363  2      RETURN 0;
;  1250          1364  1      END;


                                        03FC 00000 OPCODE_NAME_INDEX:
                                                   .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9          ; 1270
                         56       04    AC  D0 00002       MOVL    INPUT_DESC, R6               ; 1315
                         01             6C  91 00006       CMPB    (AP), #1                     ; 1313
                                        0F  1B 00009       BLEQU   1$                           ;
                                     08 AC  DD 0000B       PUSHL   DELIMITER                    ; 1315
                                        56  DD 0000E       PUSHL   R6                           ;
                   0000V CF             02  FB 00010       CALLS   #2, SCAN_OPERAND             ;
                         59             50  D0 00015       MOVL    R0, LENGTH                   ;
                                        31  11 00018       BRB     5$                           ;
                                        56  DD 0001A 1$:    PUSHL   R6                           ; 1321
                   0000V CF             01  FB 0001C       CALLS   #1, SKIP_LEADING_BLANKS      ;
                         51       04    A6  D0 00021       MOVL    4(R6), TEMP_PTR              ; 1323
                                        59  D4 00025       CLRL    LENGTH                       ; 1328
```

```
   59        66           10              00  ED 00027 2$:     CMPZV   #0, #16, (R6), LENGTH                          ; 1329
                                          1D  15 0002C         BLEQ    5$
                          50              81  90 0002E         MOVB    (TEMP_PTR)+, CHAR                              ; 1332
                     41   8F              50  91 00031         CMPB    CHAR, #65                                      ; 1333
                                          06  1F 00035         BLSSU   3$
                     5A   8F              50  91 00037         CMPB    CHAR, #90                                      ;
                                          0A  1B 0003B         BLEQU   4$
                          30              50  91 0003D 3$:     CMPB    CHAR, #48                                      ; 1334
                                          09  1F 00040         BLSSU   5$
                          39              50  91 00042         CMPB    CHAR, #57                                      ;
                                          04  1A 00045         BGTRU   5$
                                          59  D6 00047 4$:     INCL    LENGTH                                         ; 1336
                                          DC  11 00049         BRB     2$
                                          55  D4 0004B 5$:     CLRL    LIMIT_LSS                                      ; 1342
                          58       0143   8F  3C 0004D         MOVZWL  #323, LIMIT_GTR                                ; 1343
             50           55              58  C1 00052 6$:     ADDL3   LIMIT_GTR, LIMIT_LSS, R0                       ; 1347
             54           50              02  C7 00056         DIVL3   #2, R0, INDEX
                          55              54  D1 0005A         CMPL    INDEX, LIMIT_LSS                               ;
                                          3F  13 0005D         BEQL    10$
             50           54              0A  C5 0005F         MULL3   #10, INDEX, R0                                 ; 1349
                          57              01  D0 00063         MOVL    #1, R7
   59        20 00000000'EF40             06  2D 00066         CMPC5   #6, DBG$OPCODE_NAME_TABLE[R0], #32, -          ;
                                   04     B6     00070                 LENGTH, @4(R6)
                                          03  1A 00072         BGTRU   7$
                          57              01  D9 00074         SBWC    #1, R7
                FFFFFFFF  8F              57  D1 00077 7$:     CMPL    R7, #-1                                        ; 1352
                                          05  12 0007E         BNEQ    8$
                          55              54  D0 00080         MOVL    INDEX, LIMIT_LSS                               ;
                                          CD  11 00083         BRB     6$
                          01              57  D1 00085 8$:     CMPL    R7, #1                                         ; 1353
                                          05  12 00088         BNEQ    9$
                          58              54  D0 0008A         MOVL    INDEX, LIMIT_GTR                               ;
                                          C3  11 0008D         BRB     6$
                                          57  D5 0008F 9$:     TSTL    R7                                             ; 1354
                                          BF  12 00091         BNEQ    6$
                          66              59  A2 00093         SUBW2   LENGTH, (R6)                                   ; 1356
                     04   A6              59  C0 00096         ADDL2   LENGTH, 4(R6)                                  ; 1357
                          50              54  D0 0009A         MOVL    INDEX, R0                                      ; 1358
                                          04     0009D         RET
                                   04     A6  DD 0009E 10$:    PUSHL   4(R6)                                          ; 1362
                                          59  DD 000A1         PUSHL   LENGTH
                                          02  DD 000A3         PUSHL   #2
                               00028208   8F  DD 000A5         PUSHL   #164360
                00000000G 00              04  FB 000AB         CALLS   #4, LIB$SIGNAL                                 ;
                                          50  D4 000B2         CLRL    R0                                             ; 1363
                                          04     000B4         RET                                                    ; 1364

; Routine Size:  181 bytes,    Routine Base:  DBG$CODE + 0806
```

```
; 1252          1365  1 ROUTINE Fetch_Instruction(Pointer,Context) : NOVALUE =
; 1253          1366  2     BEGIN
; 1254          1367  2     BUILTIN PROBER,MOVC5;
; 1255          1368  2
; 1256          1369  2     IF NOT PROBER(%REF(0),%REF(.Data_Size[.Context]),..Pointer)
; 1257          1370  2      THEN SIGNAL(dbg$_noaccessr,1,..Pointer);
; 1258          1371  2     MOVC5(%REF(.Data_Size[.Context]),..Pointer,%REF(0),
; 1259          1372  2          %REF(16),Op_Buffer;,.Pointer);
; 1260          1373  2     SELECTONE .Context OF
; 1261          1374  2         SET
; 1262          1375  2         [context_b]:     Op_Buffer[s_long] = .Op_Buffer[s_byte];
; 1263          1376  2         [context_w]:     Op_Buffer[s_long] = .Op_Buffer[s_word];
; 1264          1377  2         [OTHERWISE]:     0;
; 1265          1378  2         TES;
; 1266          1379  1     END;



                                                   03FC 00000 FETCH_INSTRUCTION:
                                                                  .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9                  ; 1365
                             59 00000000'  EF   9E 00002          MOVAB   DATA_SIZE, R9                                  ;
                             58 00000000'  EF   9E 00009          MOVAB   OP_BUFFER, R8                                  ;
                             56          08  AC   D0 00010          MOVL    CONTEXT, R6                                    ; 1369
                             50        6946       9A 00014          MOVZBL  DATA_SIZE[R6], R0                              ;
                             57          04  AC   D0 00018          MOVL    POINTER, R7                                    ;
              00    B7       50              00   0C 0001C          PROBER  #0, R0, @0(R7)                                 ;
                                             11   12 00021          BNEQ    1$                                             ;
                                             67   DD 00023          PUSHL   (R7)                                           ; 1370
                                             01   DD 00025          PUSHL   #1                                             ;
                                 00028228    8F   DD 00027          PUSHL   #164392                                        ;
                      00000000G  00          03   FB 0002D          CALLS   #3, LIB$SIGNAL                                 ;
                             50        6946       9A 00034 1$:      MOVZBL  DATA_SIZE[R6], R0                              ; 1371
         10          00         00   B7      50   2C 00038          MOVC5   R0, @0(R7), #0, #16, OP_BUFFER                 ;
                                             68      0003E                                                                 ;
                             67              51   D0 0003F          MOVL    R1, (R7)                                       ; 1372
                                             56   D5 00042          TSTL    R6                                             ; 1375
                                             04   12 00044          BNEQ    2$                                             ;
                             68              68   98 00046          CVTBL   OP_BUFFER, OP_BUFFER                           ;
                                                  04 00049          RET                                                    ;
                             01              56   D1 0004A 2$:      CMPL    R6, #1                                         ; 1376
                                             03   12 0004D          BNEQ    3$                                             ;
                             68              68   32 0004F          CVTWL   OP_BUFFER, OP_BUFFER                           ;
                                                  04 00052 3$:      RET                                                    ; 1379

; Routine Size:  83 bytes,    Routine Base:  DBG$CODE + 08BB
```

```
;  1268          1380  1 ROUTINE Fetch_Operand(Pointer,Context,Print_Flag,Index_Flag) : NOVALUE =
;  1269          1381  2     BEGIN
;  1270          1382  2     LOCAL
;  1271          1383  2         Mode_Specifier,
;  1272          1384  2         Register_Field;
;  1273          1385  2
;  1274          1386  2
;  1275          1387  2     SELECTONE .Context OF
;  1276          1388  2         SET
;  1277          1389  2         [context_p,context_t]:
;  1278          1390  3             BEGIN
;  1279          1391  3             Fetch_Operand(.Pointer,context_wu,.Print_Flag,.Index_Flag);
;  1280          1392  3             IF .Print_Flag THEN DBG$Print(Format_AD,1,comma);
;  1281          1393  3             Fetch_Operand(.Pointer,context_b,.Print_Flag,.Index_Flag);
;  1282          1394  2             END;
;  1283          1395  2         [context_m,context_v]:
;  1284          1396  3             BEGIN
;  1285          1397  3             Fetch_Operand(.Pointer,context_l,.Print_Flag,.Index_Flag);
;  1286          1398  3             IF .Print_Flag THEN DBG$Print(Format_AD,1,comma);
;  1287          1399  3             Fetch_Operand(.Pointer,context_b,.Print_Flag,.Index_Flag);
;  1288          1400  3             IF .context EQL context_v THEN RETURN;
;  1289          1401  3             IF .Print_Flag THEN DBG$Print(Format_AD,1,comma);
;  1290          1402  3             Fetch_Operand(.Pointer,context_b,.Print_Flag,.Index_Flag);
;  1291          1403  2             END;
;  1292          1404  2         [OTHERWISE]:
;  1293          1405  3             BEGIN
;  1294          1406  3             Fetch_Instruction(.Pointer,context_bu);
;  1295          1407  3             Mode_Specifier = .Op_Buffer[0,4,4,0];
;  1296          1408  3             Register_Field = .Op_Buffer[0,0,4,0];
;  1297          1409  3
;  1298          1410  4             IF (.Print_Flag AND .Index_Flag AND (.Mode_Specifier LEQ 5))
;  1299          1411  3              THEN SIGNAL(dbg$_addressmode);
;  1300          1412  3
;  1301          1413  3             CASE .Mode_Specifier FROM 0 TO 15 OF
;  1302          1414  3                 SET
;  1303          1415  3                 [0,1,2,3]:      ! Literal
;  1304          1416  3                   IF (.Print_Flag) THEN
;  1305          1417  4                     BEGIN
;  1306          1418  4                     DBG$Print(Format_AD,3,UPLIT BYTE('S^#'));
;  1307          1419  4                     SELECTONE .Context OF
;  1308          1420  4                         SET
;  1309          1421  4                         [context_f,context_d,
;  1310          1422  4                          context_g,context_h]:
;  1311          1423  5                             BEGIN
;  1312          1424  5                             Operand_Value = (.Operand_Value^4) OR %X'4000';
;  1313          1425  5                             Print_Operand(context_f);
;  1314          1426  4                             END;
;  1315          1427  4                         [OTHERWISE]:
;  1316          1428  5                             BEGIN
;  1317          1429  5                             Print_Operand(Context_b);
;  1318          1430  4                             END;
;  1319          1431  4                     TES;
;  1320          1432  3                     END;
;  1321          1433  3
;  1322          1434  3                 [4,5,6,7,8,9]:  ! Various Register Modes
;  1323          1435  4                     BEGIN
;  1324          1436  5                     IF (.Mode_Specifier<3,1,0> AND (.Register_Field EQL 15))
```

```
;  1325          1437  4                THEN
;  1326          1438  5                    BEGIN
;  1327          1439  5                    IF NOT .Mode_Specifier
;  1328          1440  5                    THEN
;  1329          1441  6                        BEGIN
;  1330          1442  6                        Fetch_Instruction(.Pointer,.Context);
;  1331          1443  6                        IF .Print_Flag THEN
;  1332          1444  7                            BEGIN
;  1333          1445  7                            DBG$Print(Format_AD,3,UPLIT BYTE('I^#'));
;  1334          1446  7                            Print_Operand(.Context);
;  1335          1447  6                            END;
;  1336          1448  6                        END
;  1337          1449  5                    ELSE
;  1338          1450  6                        BEGIN
;  1339          1451  6                        Fetch_Instruction(.Pointer,context_l);
;  1340          1452  6                        IF .Print_Flag THEN
;  1341          1453  7                            BEGIN
;  1342          1454  7                            DBG$Print(Format_AD,2,UPLIT BYTE('@#'));
;  1343          1455  7                            Print_Address(.Operand_Value);
;  1344          1456  6                            END;
;  1345          1457  6                        END
;  1346          1458  5                    END
;  1347          1459  4                ELSE
;  1348          1460  5                    BEGIN
;  1349          1461  5                    IF .Mode_Specifier EQL 4 THEN
;  1350          1462  5                        Fetch_Operand(.Pointer,.Context,.Print_Flag,1);
;  1351          1463  5                    IF .Print_Flag THEN
;  1352          1464  6                        BEGIN
;  1353          1465  6                        BIND Punctuation = UPLIT BYTE(
;  1354          1466  6                             0 ,'[','],' 0 ,          ! Mode 4
;  1355          1467  6                             0 , 0 , 0 , 0 ,          ! Mode 5
;  1356          1468  6                             0 ,'(',')', 0 ,          ! Mode 6
;  1357          1469  6                            '-','(',')', 0 ,          ! Mode 7
;  1358          1470  6                             0 ,'(',')','+',          ! Mode 8
;  1359          1471  6                            '@','(',')','+')          ! Mode 9
;  1360          1472  6                            - (4*4) : BLOCKVECTOR [10,4,BYTE];
;  1361          1473  6
;  1362          1474  6                        IF .Punctuation[.Mode_Specifier,0,0,8,0] NEQ 0 THEN
;  1363          1475  6                            DBG$Print(Format_AD,1,Punctuation[.Mode_Specifier,0,0,0,0]);
;  1364          1476  6                        IF .Punctuation[.Mode_Specifier,1,0,8,0] NEQ 0 THEN
;  1365          1477  6                            DBG$Print(Format_AD,1,Punctuation[.Mode_Specifier,1,0,0,0]);
;  1366          1478  6                        DBG$Print(Format_AC,Register_Name[,Register_Field]);
;  1367          1479  6                        IF .Punctuation[.Mode_Specifier,2,0,8,0] NEQ 0 THEN
;  1368          1480  6                            DBG$Print(Format_AD,1,Punctuation[.Mode_Specifier,2,0,0,0]);
;  1369          1481  6                        IF .Punctuation[.Mode_Specifier,3,0,8,0] NEQ 0 THEN
;  1370          1482  6                            DBG$Print(Format_AD,1,Punctuation[.Mode_Specifier,3,0,0,0]);
;  1371          1483  5                        END;
;  1372          1484  4                    END;
;  1373          1485  3                END;
;  1374          1486  3
;  1375          1487  3            [10,11,12,13,14,15]:    ! Displacement Modes
;  1376          1488  4                BEGIN
;  1377          1489  4                LOCAL Offset_Context;
;  1378          1490  4
;  1379          1491  4                Offset_Context = (.Mode_Specifier-10)/2;
;  1380          1492  4                Fetch_Instruction(.Pointer,.Offset_Context);
;  1381          1493  4                IF (.Print_Flag) THEN
```

```
;  1382          1494  5                     BEGIN
;  1383          1495  5                     BIND Mode_Char = UPLIT BYTE('BWL') : VECTOR[,BYTE];
;  1384          1496  5                     IF (.Mode_Specifier) THEN
;  1385          1497  5                         DBG$Print(Format_AD,1,UPLIT BYTE('@'));
;  1386          1498  5                     DBG$Print(Format_AD,1,Mode_Char[.Offset_Context]);
;  1387          1499  5                     DBG$Print(Format_AD,1,UPLIT BYTE('^'));
;  1388          1500  6                     IF (.Register_Field EQL 15)
;  1389          1501  5                     THEN
;  1390          1502  5                         Print_Address(.Operand_Value + ..Pointer)
;  1391          1503  5                     ELSE
;  1392          1504  6                         BEGIN
;  1393          1505  6                         Print_Operand(.Offset_Context);
;  1394          1506  6                         DBG$Print(Format_AD,1,UPLIT BYTE('('));
;  1395          1507  6                         DBG$Print(Format_AC,Register_Name[,Register_Field]);
;  1396          1508  6                         DBG$Print(Format_AD,1,UPLIT BYTE(')'));
;  1397          1509  5                         END;
;  1398          1510  4                     END;
;  1399          1511  3                 END;
;  1400          1512  3             TES;
;  1401          1513  2         END;
;  1402          1514  2     TES;
;  1403          1515  1 END;


                                                .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

                                        23  5E  53  01178 P.AAK:   .ASCII  \S^#\                                    ;
                                        23  5E  49  0117B P.AAL:   .ASCII  \I^#\                                    ;
                                            23  40  0117E P.AAM:   .ASCII  \@#\                                    ;
                                                00  01180 P.AAN:   .BYTE   0                                       ;
                                                5B  01181          .ASCII  \[\                                     ;
                                                5D  01182          .ASCII  \]\                                     ;
                                    00  00  00  00  00  00  01183          .BYTE   0, 0, 0, 0, 0, 0                ;
                                                28  01189          .ASCII  \(\                                     ;
                                                29  0118A          .ASCII  \)\                                     ;
                                                00  0118B          .BYTE   0                                       ;
                                                2D  0118C          .ASCII  \-\                                     ;
                                                28  0118D          .ASCII  \(\                                     ;
                                                29  0118E          .ASCII  \)\                                     ;
                                                    00  0118F          .BYTE   0, 0                                ;
                                                28  01191          .ASCII  \(\                                     ;
                                                29  01192          .ASCII  \)\                                     ;
                                                2B  01193          .ASCII  \+\                                     ;
                                                40  01194          .ASCII  \@\                                     ;
                                                28  01195          .ASCII  \(\                                     ;
                                                29  01196          .ASCII  \)\                                     ;
                                                2B  01197          .ASCII  \+\                                     ;
                                        4C  57  42  01198 P.AAO:   .ASCII  \BWL\                                   ;
                                                40  0119B P.AAP:   .ASCII  \@\                                     ;
                                                5E  0119C P.AAQ:   .ASCII  \^\                                     ;
                                                28  0119D P.AAR:   .ASCII  \(\                                     ;
                                                29  0119E P.AAS:   .ASCII  \)\                                     ;

                                                        PUNCTUATION=        P.AAN-16
                                                        MODE_CHAR=          P.AAO
```

```
                                        .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                        07FC 00000 FETCH_OPERAND:
                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10        ; 1380
5A          FB  AF  9E 00002            MOVAB   FETCH_OPERAND, R10
59 00000000'    EF  9E 00006            MOVAB   OP_BUFFER, R9
58 00000000G    00  9E 0000D            MOVAB   DBG$PRINT, R8
57 00000000'    EF  9E 00014            MOVAB   FORMAT_AD, R7
54          08  AC  D0 0001B            MOVL    CONTEXT, R4                             ; 1387
0B              54  D1 0001F            CMPL    R4, #11                                 ; 1389
                25  19 00022            BLSS    2$
0C              54  D1 00024            CMPL    R4, #12
                20  14 00027            BGTR    2$
7E          0C  AC  7D 00029            MOVQ    PRINT_FLAG, -(SP)                       ; 1391
                0A  DD 0002D            PUSHL   #10
            04  AC  DD 0002F            PUSHL   POINTER
6A              04  FB 00032            CALLS   #4, FETCH_OPERAND
0A          0C  AC  E9 00035            BLBC    PRINT_FLAG, 1$                          ; 1392
            08  A7  9F 00039            PUSHAB  COMMA
                01  DD 0003C            PUSHL   #1
                57  DD 0003E            PUSHL   R7
68              03  FB 00040            CALLS   #3, DBG$PRINT
7E          0C  AC  7D 00043 1$:        MOVQ    PRINT_FLAG, -(SP)                       ; 1393
                4D  11 00047            BRB     6$
0D              54  D1 00049 2$:        CMPL    R4, #13                                 ; 1395
                51  19 0004C            BLSS    7$
0E              54  D1 0004E            CMPL    R4, #14
                4C  14 00051            BGTR    7$
            10  AC  DD 00053            PUSHL   INDEX_FLAG                              ; 1397
52          0C  AC  D0 00056            MOVL    PRINT_FLAG, R2
                52  DD 0005A            PUSHL   R2
                02  DD 0005C            PUSHL   #2
            04  AC  DD 0005E            PUSHL   POINTER
6A              04  FB 00061            CALLS   #4, FETCH_OPERAND
0A              52  E9 00064            BLBC    R2, 3$                                  ; 1398
            08  A7  9F 00067            PUSHAB  COMMA
                01  DD 0006A            PUSHL   #1
                57  DD 0006C            PUSHL   R7
68              03  FB 0006E            CALLS   #3, DBG$PRINT
            10  AC  DD 00071 3$:        PUSHL   INDEX_FLAG                              ; 1399
                52  DD 00074            PUSHL   R2
                7E  D4 00076            CLRL    -(SP)
            04  AC  DD 00078            PUSHL   POINTER
6A              04  FB 0007B            CALLS   #4, FETCH_OPERAND
0E              54  D1 0007E            CMPL    R4, #14                                 ; 1400
                01  12 00081            BNEQ    4$
                    04 00083            RET
0A              52  E9 00084 4$:        BLBC    R2, 5$                                  ; 1401
            08  A7  9F 00087            PUSHAB  COMMA
                01  DD 0008A            PUSHL   #1
                57  DD 0008C            PUSHL   R7
68              03  FB 0008E            CALLS   #3, DBG$PRINT
            10  AC  DD 00091 5$:        PUSHL   INDEX_FLAG                              ; 1402
                52  DD 00094            PUSHL   R2
                7E  D4 00096 6$:        CLRL    -(SP)
            04  AC  DD 00098            PUSHL   POINTER
```

```
                          6A            04  FB 0009B        CALLS   #4, FETCH_OPERAND                        ;
                                        04  0009E           RET                                              ; 1387
                          09            DD  0009F 7$:       PUSHL   #9                                       ; 1406
                          55      04    AC  D0 000A1        MOVL    POINTER, R5                              ;
                                        55  DD 000A5        PUSHL   R5                                       ;
                    AD    AA            02  FB 000A7        CALLS   #2, FETCH_INSTRUCTION                    ;
    52          69        04            04  EF 000AB        EXTZV   #4, #4, OP_BUFFER, MODE_SPECIFIER        ; 1407
    53          69        04            00  EF 000B0        EXTZV   #0, #4, OP_BUFFER, REGISTER_FIELD        ; 1408
                          56      0C    AC  D0 000B5        MOVL    PRINT_FLAG, R6                           ; 1410
                          16            56  E9 000B9        BLBC    R6, 8$                                   ;
                          12      10    AC  E9 000BC        BLBC    INDEX_FLAG, 8$                           ;
                          05            52  D1 000C0        CMPL    MODE_SPECIFIER, #5                       ;
                                        0D  14 000C3        BGTR    8$                                       ;
                              00028753  8F  DD 000C5        PUSHL   #165715                                  ; 1411
                00000000G 00            01  FB 000CB        CALLS   #1, LIB$SIGNAL                           ;
                0F        00            52  CF 000D2 8$:    CASEL   MODE_SPECIFIER, #0, #15                  ; 1413
  0020          0020      0020          0020   000D6 9$:    .WORD   10$-9$,-                                 ;
  004B          004B      004B          004B   000DE                10$-9$,-                                 ;
  00F2          00F2      004B          004B   000E6                10$-9$,-                                 ;
  00F2          00F2      00F2          00F2   000EE                10$-9$,-                                 ;
                                                                    12$-9$,-                                 ;
                                                                    12$-9$,-                                 ;
                                                                    12$-9$,-                                 ;
                                                                    12$-9$,-                                 ;
                                                                    12$-9$,-                                 ;
                                                                    12$-9$,-                                 ;
                                                                    20$-9$,-                                 ;
                                                                    20$-9$,-                                 ;
                                                                    20$-9$,-                                 ;
                                                                    20$-9$,-                                 ;
                                                                    20$-9$,-                                 ;
                                                                    20$-9$                                   ;
                          7B            56  E9 000F6 10$:   BLBC    R6, 16$                                  ; 1416
                                  70    A7  9F 000F9        PUSHAB  P.AAK                                    ; 1418
                                        03  DD 000FC        PUSHL   #3                                       ;
                                        57  DD 000FE        PUSHL   R7                                       ;
                          68            03  FB 00100        CALLS   #3, DBG$PRINT                            ;
                          05            54  D1 00103        CMPL    R4, #5                                   ; 1421
                                        15  19 00106        BLSS    11$                                      ;
                          08            54  D1 00108        CMPL    R4, #8                                   ;
                                        10  14 0010B        BGTR    11$                                      ;
                50        69            04  78 0010D        ASHL    #4, OPERAND_VALUE, R0                    ; 1424
                69        50  00004000  8F  C9 00111        BISL3   #16384, R0, OPERAND_VALUE                ;
                                        05  DD 00119        PUSHL   #5                                       ; 1425
                                        27  11 0011B        BRB     13$                                      ;
                                        7E  D4 0011D 11$:   CLRL    -(SP)                                    ; 1429
                                        23  11 0011F        BRB     13$                                      ;
                3F        52            03  E1 00121 12$:   BBC     #3, MODE_SPECIFIER, 15$                  ; 1436
                          0F            53  D1 00125        CMPL    REGISTER_FIELD, #15                      ;
                                        3A  12 00128        BNEQ    15$                                      ;
                          1D            52  E8 0012A        BLBS    MODE_SPECIFIER, 14$                      ; 1439
                                        54  DD 0012D        PUSHL   R4                                       ; 1442
                                        55  DD 0012F        PUSHL   R5                                       ;
                    AD    AA            02  FB 00131        CALLS   #2, FETCH_INSTRUCTION                    ;
                          3C            56  E9 00135        BLBC    R6, 16$                                  ; 1443
                                  73    A7  9F 00138        PUSHAB  P.AAL                                    ; 1445
                                        03  DD 0013B        PUSHL   #3                                       ;
```

```
                        57  DD 0013D          PUSHL   R7
                    68  03  FB 0013F          CALLS   #3, DBG$PRINT                                   ;
                        54  DD 00142          PUSHL   R4                                              ; 1446
                0000V   CF  01  FB 00144 13$:  CALLS   #1, PRINT_OPERAND                               ;
                            04 00149          RET                                                     ; 1438
                        02  DD 0014A 14$:      PUSHL   #2                                              ; 1451
                        55  DD 0014C          PUSHL   R5                                              ;
                  AD    AA  02  FB 0014E      CALLS   #2, FETCH_INSTRUCTION                           ;
                        1F  56  E9 00152      BLBC    R6, 16$                                         ; 1452
                          76    A7  9F 00155  PUSHAB  P.AAM                                           ; 1454
                            02  DD 00158      PUSHL   #2                                              ;
                            57  DD 0015A      PUSHL   R7                                              ;
                    68          03  FB 0015C  CALLS   #3, DBG$PRINT                                   ;
                            69  DD 0015F      PUSHL   OPERAND_VALUE                                   ; 1455
                        00A5    31  00161      BRW     23$                                             ;
                    04          52  D1 00164 15$:  CMPL    MODE_SPECIFIER, #4                      ; 1461
                            0B  12 00167      BNEQ    16$                                             ;
                            01  DD 00169      PUSHL   #1                                              ; 1462
                        0050    8F  BB 0016B  PUSHR   #^M<R4,R6>                                      ;
                            55  DD 0016F      PUSHL   R5                                              ;
                    6A          04  FB 00171  CALLS   #4, FETCH_OPERAND                               ;
                    61          56  E9 00174 16$:  BLBC    R6, 21$                                 ; 1463
                          68 A742   DF 00177  PUSHAL  PUNCTUATION[MODE_SPECIFIER]                     ; 1474
                            9E  95 0017B      TSTB    @(SP)+                                          ;
                            0B  13 0017D      BEQL    17$                                             ;
                          68 A742   DF 0017F  PUSHAL  PUNCTUATION[MODE_SPECIFIER]                     ; 1475
                            01  DD 00183      PUSHL   #1                                              ;
                            57  DD 00185      PUSHL   R7                                              ;
                    68          03  FB 00187  CALLS   #3, DBG$PRINT                                   ;
                          69 A742   DF 0018A 17$:  PUSHAL  PUNCTUATION+1[MODE_SPECIFIER]              ; 1476
                            9E  95 0018E      TSTB    @(SP)+                                          ;
                            0B  13 00190      BEQL    18$                                             ;
                          69 A742   DF 00192  PUSHAL  PUNCTUATION+1[MODE_SPECIFIER]                   ; 1477
                            01  DD 00196      PUSHL   #1                                              ;
                            57  DD 00198      PUSHL   R7                                              ;
                    68          03  FB 0019A  CALLS   #3, DBG$PRINT                                   ;
                        B0  A743    DF 0019D 18$:  PUSHAL  REGISTER_NAME[REGISTER_FIELD]              ; 1478
                        04    A7    9F 001A1  PUSHAB  FORMAT_AC                                       ;
                    68          02  FB 001A4  CALLS   #2, DBG$PRINT                                   ;
                          6A A742   DF 001A7  PUSHAL  PUNCTUATION+2[MODE_SPECIFIER]                   ; 1479
                            9E  95 001AB      TSTB    @(SP)+                                          ;
                            0B  13 001AD      BEQL    19$                                             ;
                          6A A742   DF 001AF  PUSHAL  PUNCTUATION+2[MODE_SPECIFIER]                   ; 1480
                            01  DD 001B3      PUSHL   #1                                              ;
                            57  DD 001B5      PUSHL   R7                                              ;
                    68          03  FB 001B7  CALLS   #3, DBG$PRINT                                   ;
                          6B A742   DF 001BA 19$:  PUSHAL  PUNCTUATION+3[MODE_SPECIFIER]              ; 1481
                            9E  95 001BE      TSTB    @(SP)+                                          ;
                            74  13 001C0      BEQL    26$                                             ;
                          6B A742   DF 001C2  PUSHAL  PUNCTUATION+3[MODE_SPECIFIER]                   ; 1482
                            67  11 001C6      BRB     25$                                             ;
                    50          F6    A2 9E 001C8 20$:  MOVAB   -10(R2), R0                         ; 1491
                54  50          02  C7 001CC  DIVL3   #2, R0, OFFSET_CONTEXT                          ;
                            54  DD 001D0      PUSHL   OFFSET_CONTEXT                                  ; 1492
                            55  DD 001D2      PUSHL   R5                                              ;
                  AD    AA      02  FB 001D4  CALLS   #2, FETCH_INSTRUCTION                           ;
                        5B      56  E9 001D8 21$:  BLBC    R6, 26$                                 ; 1493
```

```
                         0B                 52  E9 001DB             BLBC    MODE_SPECIFIER, 22$                     ; 1496
                                      0093  C7  9F 001DE             PUSHAB  P.AAP                                   ; 1497
                                            01  DD 001E2             PUSHL   #1
                                            57  DD 001E4             PUSHL   R7
                         68                 03  FB 001E6             CALLS   #3, DBG$PRINT
                                      0090 C744 9F 001E9 22$:         PUSHAB  MODE_CHAR[OFFSET_CONTEXT]               ; 1498
                                            01  DD 001EE             PUSHL   #1
                                            57  DD 001F0             PUSHL   R7
                         68                 03  FB 001F2             CALLS   #3, DBG$PRINT
                                      0094  C7  9F 001F5             PUSHAB  P.AAQ                                   ; 1499
                                            01  DD 001F9             PUSHL   #1
                                            57  DD 001FB             PUSHL   R7
                         68                 03  FB 001FD             CALLS   #3, DBG$PRINT
                         0F                 53  D1 00200             CMPL    REGISTER_FIELD, #15                     ; 1500
                                            0A  12 00203             BNEQ    24$
                      7E                    69  C1 00205             ADDL3   (R5), OPERAND_VALUE, -(SP)              ; 1502
                                0000V       CF  01 FB 00209 23$:     CALLS   #1, PRINT_ADDRESS
                                            04  0020E               RET
                                            54  DD 0020F 24$:        PUSHL   OFFSET_CONTEXT                          ; 1505
                                0000V       CF  01 FB 00211          CALLS   #1, PRINT_OPERAND
                                      0095  C7  9F 00216             PUSHAB  P.AAR                                   ; 1506
                                            01  DD 0021A             PUSHL   #1
                                            57  DD 0021C             PUSHL   R7
                         68                 03  FB 0021E             CALLS   #3, DBG$PRINT
                                      B0 A743 DF 00221               PUSHAL  REGISTER_NAME[REGISTER_FIELD]           ; 1507
                                      04  A7  9F 00225               PUSHAB  FORMAT_AC
                         68                 02  FB 00228             CALLS   #2, DBG$PRINT
                                      0096  C7  9F 0022B             PUSHAB  P.AAS                                   ; 1508
                                            01  DD 0022F 25$:        PUSHL   #1
                                            57  DD 00231             PUSHL   R7
                         68                 03  FB 00233             CALLS   #3, DBG$PRINT
                                            04  00236 26$:           RET                                             ; 1515

; Routine Size:  567 bytes,    Routine Base:  DBG$CODE + 090E
```

```
;  1405          1516  1 ROUTINE Parse_Operand(EnCode,Context,Operand_number) : NOVALUE =                  ! M007
;  1406          1517  2     BEGIN
;  1407          1518  2     MAP EnCode  : REF BLOCK [20,BYTE] FIELD(EnCode_Fields);
;  1408          1519  2     LABEL scan,trim,mode,Addr;
;  1409          1520  2     LOCAL
;  1410          1521  2         Regnum,R_Mode,  ! Index register and addressing mode
;  1411          1522  2         Indexed_Regnum,                                          ! For saving the indexed regi
;  1412          1523  2         Displacement_size_needed,                                ! Displacement sized needed f
;  1413          1524  2         Length,Parsed,  ! Operand lengths (unparsed and total)
;  1414          1525  2         Defer,Address,  ! Operand Mode and Address
;  1415          1526  2         Buffer          : REF VECTOR [,BYTE];
;  1416          1527  2
;  1417          1528  2     MACRO Report_Error = (SIGNAL(dbg$_opsyntax,1,..Operand_number);RETURN)%; ! saves Code M007
;  1418          1529  2
;  1419          1530  2     SELECTONE .Context OF
;  1420          1531  2         SET
;  1421          1532  2         [context_p,context_t]:
;  1422          1533  3             BEGIN
;  1423          1534  3             Parse_Operand(.EnCode,context_wu,.Operand_number);   ! M007
;  1424          1535  3             Parse_Operand(.EnCode,context_b,.Operand_number);    ! M007
;  1425          1536  3             RETURN;
;  1426          1537  3             END;
;  1427          1538  2         [context_m,context_v]:
;  1428          1539  3             BEGIN
;  1429          1540  3             Parse_Operand(.EnCode,context_l,.Operand_number);    ! M007
;  1430          1541  3             Parse_Operand(.EnCode,context_b,.Operand_number);    ! M007
;  1431          1542  3             IF .context EQL context_m THEN Parse_Operand(.EnCode,context_b,.Operand_number);! M007
;  1432          1543  3             RETURN;
;  1433          1544  3             END;
;  1434          1545  2
;  1435          1546  2         [OTHERWISE]:
;  1436          1547  3             BEGIN
;  1437          1548  3             .Operand_number = ..Operand_number + 1;                  ! A007
;  1438          1549  3             Length = Parsed = Scan_Operand(EnCode[Enc_Input_Desc],%C',');
;  1439          1550  3             Buffer = .EnCode[Enc_Input_Buffer];
;  1440          1551  3             WHILE (.Length GTR 0) AND (.Buffer[.Length-1] EQL %C' ') DO Length = .Length - 1;
;  1441          1552  3             IF .Length LEQ 0 THEN SIGNAL(DBG$_INCOMPOPR, 1, ..Operand_number);  ! M007
;  1442          1553  3             END;
;  1443          1554  2         TES;
;  1444          1555  2
;  1445          1556  2
```

```
;  1447          1557  3 scan:   BEGIN
;  1448          1558  3 !++
;  1449          1559  3 ! Trim block takes the -(Rn)[Rn] off the operand
;  1450          1560  3 ! starting at the back and trimming toward the front
;  1451          1561  3 !--
;  1452          1562  4 trim:       BEGIN
;  1453          1563  4             R_Mode = 0;
;  1454          1564  4             Indexed_Regnum = -1;                                        ! No indexed register seen ye
;  1455          1565  4             Displacement_size_needed = 0;                               ! No displacement size needed
;  1456          1566  4             IF .Buffer[.Length-1] EQL %C']' THEN                        ! Index mode?
;  1457          1567  5                 BEGIN
;  1458          1568  5                 DECR Start FROM .Length-1 TO 1 DO                       ! Look for matching [
;  1459          1569  5                   IF .Buffer[.Start-1] EQL %C'[' THEN
;  1460          1570  6                     BEGIN
;  1461          1571  6                     Regnum = Parse_Register(Buffer[.Start],.Length-1-.Start);  ! Get the register number
;  1462          1572  6                     IF .Regnum LSS 0                                    ! Bad register, Assume addr-e
;  1463          1573  6                     THEN
;  1464          1574  6                         SIGNAL(DBG$_REGREQ, 1, ..Operand_number);       !
;  1465          1575  6                     IF .Regnum GEQ 15                                   ! Disallow PC
;  1466          1576  6                     THEN
;  1467          1577  6                         SIGNAL(DBG$_PCNOTALL, 1, ..Operand_number);     !
;  1468          1578  6                     IF (Length = .Start - 1) LEQ 0                      ! Consume the [Rx]
;  1469          1579  6                     THEN
;  1470          1580  6                         SIGNAL(DBG$_INCOMPOPR, 1, ..Operand_number);    ! make sure there is more
;  1471          1581  6                     Indexed_Regnum = .Regnum;                           ! Save the indexed register n
;  1472          1582  6                     Regnum = .Regnum + %X'40';                          ! Construct the index mode by
;  1473          1583  6                     Store_Operand(.EnCode,Regnum,1);                    ! Store the index mode byte
;  1474          1584  6                     R_Mode = -1;                                        ! Flag that we've done index
;  1475          1585  6                     EXITLOOP;
;  1476          1586  5                     END;
;  1477          1587  4                 END;
;  1478          1588  4
;  1479          1589  4             IF .Buffer[.Length-1] EQL %C'+' THEN                        ! Autoincrement?
;  1480          1590  5                 BEGIN
;  1481          1591  5                 R_Mode = %X'80';                                        ! Construct the mode field fo
;  1482          1592  5                 IF (Length = .Length - 1) LEQ 0                         ! Consume the "+" and
;  1483          1593  5                 THEN
;  1484          1594  5                     SIGNAL(DBG$_INCOMPOPR, 1, ..Operand_number);        !  make sure there is more
;  1485          1595  4                 END;
;  1486          1596  4
;  1487          1597  4             IF .Buffer[.Length-1] EQL %C')' THEN                        ! (Rn)?
;  1488          1598  5                 BEGIN
;  1489          1599  5                 DECR Start FROM .Length-1 TO 1 DO                       ! Look for matching "("
;  1490          1600  5                   IF .Buffer[.Start-1] EQL %C'(' THEN
;  1491          1601  6                     BEGIN
;  1492          1602  6                     Regnum = Parse_Register(Buffer[.Start],.Length-1-.Start);  ! Get the register number
;  1493          1603  7                     IF (.Regnum EQL 15)                                 ! Dissallow the PC
;  1494          1604  6                     THEN
;  1495          1605  6                         SIGNAL(DBG$_PCNOTALL, 1, ..Operand_number);     !
;  1496          1606  6                     IF .R_Mode GTR 0 THEN                               ! Autoincrement from above?
;  1497          1607  7                         BEGIN                                           ! Yes.  We must have a regist
;  1498          1608  7                         IF .Regnum LSS 0                                ! Bad register
;  1499          1609  7                         THEN
;  1500          1610  7                             SIGNAL(DBG$_REGREQ, 1, ..Operand_number);   !
;  1501          1611  7                         IF (.Indexed_Regnum EQL .Regnum) AND            ! The indexed and base regist
;  1502          1612  8                            (.R_Mode EQL %X'80')                         ! same if the register is aut
;  1503          1613  7                         THEN
```

```
;  1504          1614  7                    SIGNAL(DBG$_INDBASEQL, 1, ..Operand_number);                !
;  1505          1615  7                Length = .Start - 1;                                            ! Consume the (Rn)
;  1506          1616  7                LEAVE trim;                                                     ! Done trimming
;  1507          1617  6                END;
;  1508          1618  6            IF .Regnum LSS 0                                                    ! Bad register
;  1509          1619  6            THEN
;  1510          1620  6                SIGNAL(DBG$_REGREQ, 1, ..Operand_number);                       !
;  1511          1621  6            IF .Start EQL 1                                                     ! "(" is the start?
;  1512          1622  6                THEN R_Mode = %X'60'                                            ! Yes, Construct register def
;  1513          1623  7            ELSE IF (.Buffer[0] EQL %C'-') AND (.Start EQL 2)                   ! Test for autodecrement
;  1514          1624  6                THEN R_Mode = %X'70';                                           ! Construct autodecrement
;  1515          1625  6            IF .R_Mode GTR 0 THEN                                               ! Do we have something?
;  1516          1626  7                BEGIN                                                           ! Yes
;  1517          1627  7                IF (.Indexed_Regnum EQL .Regnum) AND                            ! The indexed and base regist
;  1518          1628  8                   (.R_Mode EQL %X'70')                                         ! same if the register is aut
;  1519          1629  7                THEN
;  1520          1630  7                    SIGNAL(DBG$_INDBASEQL, 1, ..Operand_number);                !
;  1521          1631  7                Regnum = .Regnum + .R_Mode;                                     ! Construct the mode and regi
;  1522          1632  7                Store_Operand(.EnCode,Regnum,1);                                ! Store the byte
;  1523          1633  7                LEAVE scan;                                                     ! Done scanning
;  1524          1634  6                END;
;  1525          1635  6            Length = .Start - 1;                                                ! Consume the (Rn)
;  1526          1636  6            R_Mode = %X'A0';    ! Byte, Word, or Long handled later             ! Assume simple displacement
;  1527          1637  6            Displacement_size_needed = 1;                                       ! We don't know the displacem
;  1528          1638  6            LEAVE trim;                                                         ! Done trimming
;  1529          1639  5            END,
;  1530          1640  5        END
;  1531          1641  4    ELSE IF .R_Mode GTR 0 THEN Report_Error;                                    ! Autoincrement and no ")"
;  1532          1642  3    END;                    ! End of block 'trim'
```

```
;  1534        1643  3          !+
;  1535        1644  3          ! We get here with Regnum and R_Mode indicating the current
;  1536        1645  3          ! state of the parse. Possible values are :-
;  1537        1646  3          !
;  1538        1647  3          !     R_Mode          Regnum
;  1539        1648  3          !       0               *       no register operands seen
;  1540        1649  3          !      -1               *       [Rn] has been generated
;  1541        1650  3          !      80               n       (Rn)+ parsed (not generated)
;  1542        1651  3          !      A0               n       (Rn) seen - displacement expected
;  1543        1652  3          !                                 (mode will be changed if needed)
;  1544        1653  3          !-
;  1545        1654  3          !
;  1546        1655  3          IF (.Length GTR 0) AND (.Buffer[0] EQL %C'@') THEN                          ! Leading "@"?
;  1547        1656  4              BEGIN                                                                   ! Yes
;  1548        1657  4              Length = .Length - 1;                                                   ! Consume it.
;  1549        1658  4              Buffer = .Buffer + 1;
;  1550        1659  4              Defer  = %X'10';                                                        ! Set defered mode
;  1551        1660  4              END
;  1552        1661  3          ELSE                                                                        ! No
;  1553        1662  3              Defer  = 0;                                                             ! Clear defered mode
;  1554        1663  3          !
;  1555        1664  3          IF .R_Mode EQL %X'80' THEN                                                  ! Autoincrement from above?
;  1556        1665  4              BEGIN                                                                   ! Yes
;  1557        1666  4              IF .Length NEQ 0 THEN SIGNAL(DBG$_OPSYNTAX, 1, ..Operand_number);      ! More in front error
;  1558        1667  4              Regnum = .Regnum + .R_Mode + .Defer;                                    ! Construct
;  1559        1668  4              Store_Operand(.EnCode,Regnum,1);                                        ! Store operand
;  1560        1669  4              LEAVE scan;                                                             ! Done
;  1561        1670  3              END;
;  1562        1671  3          !
;  1563        1672  3          !++
;  1564        1673  3          ! By now (Rn), (Rn)+, -(Rn), @(Rn)+ are done
;  1565        1674  3          !--
;  1566        1675  3          IF (.Length LEQ 0) THEN SIGNAL(DBG$_INCOMPOPR, 1, ..Operand_number);       ! There must be more.
;  1567        1676  3          !
;  1568        1677  4 Addr:    BEGIN                                                                       ! Addr-expr got caught where
;  1569        1678  4          IF (.Buffer[0] EQL %C'#') THEN                                             ! Look for literals
;  1570        1679  5              BEGIN
;  1571        1680  5              Buffer = .Buffer + 1;                                                  ! Consume the "#"
;  1572        1681  5              Length = .Length - 1;
;  1573        1682  5              IF (.Length LEQ 0) OR (.R_Mode GTR 0) THEN Report_Error;               ! There must be more and we c
;  1574        1683  5              IF (.Defer EQL 0) AND (.R_Mode LSS 0) THEN Report_Error;               ! Indexed mode not allowed wi
;  1575        1684  5              R_Mode = %X'80';                                                       ! Assume immediate mode
;  1576        1685  5              Regnum = %X'0F';
;  1577        1686  5              END
;  1578        1687  4          ELSE                                                                       ! Not a "#" in front
;  1579        1688  5              BEGIN
;  1580        1689  5              IF (.Length LSS 2) OR (.Buffer[1] NEQ %C'^') THEN                      ! Not ?^xxxxx
;  1581        1690  6                  BEGIN
;  1582        1691  7                  IF (.R_Mode NEQ 0) OR (.Defer NEQ 0)
;  1583        1692  7                    OR (Parse_Register(.Buffer,.Length) LSS 0)
;  1584        1693  6                      THEN                                                           ! We have a Relative addr
;  1585        1694  7                          BEGIN                                                      !
;  1586        1695  7                          IF .R_Mode LEQ 0                                           ! Has a (Rn) been seen?
;  1587        1696  7                          THEN                                                       ! No
;  1588        1697  8                              BEGIN                                                  !
;  1589        1698  8                              Regnum = 15;                                           ! Construct the register
;  1590        1699  8                              R_Mode = %X'A0';                                       ! Construct the mode
```

```
;  1591          1700  8                        Displacement_size_needed = 1;                          ! The displacement size must
;  1592          1701  7                        END;                                                   !
;  1593          1702  7                    LEAVE Addr;                                                ! We must have an Addr-expr
;  1594          1703  6                    END;                                                       !
;  1595          1704  6                Regnum = Parse_Register(.Buffer, .Length) + %X'50';            ! Simple register mode
;  1596          1705  6                Store_Operand(.Encode,Regnum,1);                               ! Store it
;  1597          1706  6                LEAVE scan;                                                    ! Done
;  1598          1707  6                END
;  1599          1708  5            ELSE
;  1600          1709  6 Mode:           BEGIN                                                          ! We have ?^xxxx
;  1601          1710  6                BIND Mode_Char = UPLIT BYTE('BWLGIS') : VECTOR[6,BYTE];        ! The possible ?
;  1602          1711  6                INCR Index FROM 0 TO 5 DO                                      ! Scan down the list
;  1603          1712  6                  IF (.Buffer[0] EQL .Mode_Char[.Index]) THEN                   ! Did we find one
;  1604          1713  7                    BEGIN
;  1605          1714  7                    Buffer = .Buffer + 2;                                      ! Consume the ^?
;  1606          1715  7                    Length = .Length - 2;
;  1607          1716  8                    IF (.Length LEQ 0)                                         ! There must be more
;  1608          1717  7                    THEN
;  1609          1718  7                        SIGNAL(DBG$_INCOMPOPR, 1, ..Operand_number);           !
;  1610          1719  7                    IF .Index LEQ 3 THEN                                       ! One of BWLG
;  1611          1720  8                        BEGIN
;  1612          1721  8                        IF (.Index EQL 3) AND                                  ! If G^
;  1613          1722  9                        ((.R_Mode GTR 0) OR (.Defer NEQ 0))                    ! must not have (Rn) on end o
;  1614          1723  8                            THEN Report_Error;
;  1615          1724  8                        IF (.R_Mode LEQ 0) THEN                                ! When there was no (Rn)
;  1616          1725  9                            BEGIN
;  1617          1726  9                            R_Mode = %X'A0';                                   ! Do the right thing for rela
;  1618          1727  9                            Regnum = %X'0F';                                   ! and relative defered
;  1619          1728  8                            END;
;  1620          1729  8                        R_Mode = .R_Mode + (.Index*%X'20');                    ! Construct the mode
;  1621          1730  8                        Displacement_size_needed = 0;                          ! The displacement size has b
;  1622          1731  8                        END
;  1623          1732  7                    ELSE
;  1624          1733  8                        BEGIN
;  1625          1734  9                        IF (.R_Mode GTR 0) OR (.Defer GTR 0)                   ! (Rn) and @ not allowed
;  1626          1735  8                          OR (.Buffer[0] NEQ %C'#') THEN Report_Error;         ! Must have #
;  1627          1736  8                        R_Mode = %X'80' + (.Mode_Char[.Index]^16);             ! Construct the right mode
;  1628          1737  8                        Regnum = %X'0F';                                       ! The right register
;  1629          1738  8                        Buffer = .Buffer + 1;                                  ! Consume the #
;  1630          1739  8                        Length = .Length - 1;                                  ! Wrong sign
;  1631          1740  9                        IF (.Length LEQ 0)                                     ! There must be a addr-expr t
;  1632          1741  8                        THEN
;  1633          1742  8                            SIGNAL(DBG$_INCOMPOPR, 1, ..Operand_number);       !
;  1634          1743  7                        END;
;  1635          1744  7                    LEAVE Mode;                                                ! Done
;  1636          1745  6                    END;
;  1637          1746  6                SIGNAL(DBG$_BWLGISMUS, 1, ..Operand_number);                   ! Wrong char in front of ^
;  1638          1747  5                END;                ! End of block 'mode'
;  1639          1748  4            END;
```

```
;  1641          1749  4          IF (((.R_Mode + .Regnum + .Defer) AND %X'FF') EQL %X'8F') THEN
;  1642          1750  5              BEGIN
;  1643          1751  5              !+
;  1644          1752  5              ! We get here if there is an immediate-mode operand.
;  1645          1753  5              ! We now read the value using the EXPRESSION  syntax
;  1646          1754  5              ! of the current language and convert it to the type
;  1647          1755  5              ! required by this instruction.
;  1648          1756  5              ! Short literals are also handled here. These can be
;  1649          1757  5              ! differentiated  from other immediate-mode operands
;  1650          1758  5              ! by examining the high word of R_Mode (set to %C'S'
;  1651          1759  5              ! for short literals, and 0 or %C'I' otherwise).
;  1652          1760  5              !-
;  1653          1761  5              LOCAL
;  1654          1762  5                  Val_desc_target : REF DBG$VALDESC,
;  1655          1763  5                  Val_desc_source : REF DBG$VALDESC,
;  1656          1764  5                  vms_desc_source : BLOCK[ 8, BYTE ],
;  1657          1765  5                  vms_desc_target : BLOCK[ 8, BYTE ];
;  1658          1766  5
;  1659          1767  5              IF .R_Mode<16,8,0> EQLU %C'S' THEN
;  1660          1768  5                SELECTONE .Context OF
;  1661          1769  5                  SET
;  1662          1770  5                  [context_f,context_d,
;  1663          1771  5                   context_g,context_h]:
;  1664          1772  5                          context = context_f;
;  1665          1773  5                  [OTHERWISE]:
;  1666          1774  5                          context = context_l;
;  1667          1775  5                  TES;
;  1668          1776  5
;  1669          1777  5              !++
;  1670          1778  5              ! Set up the source
;  1671          1779  5              !--
;  1672          1780  5              vms_desc_source[dsc$b_class]   =  dsc$k_class_s;                    ! A008
;  1673          1781  5              vms_desc_source[dsc$b_dtype]   = DSC$K_DTYPE_T;                     ! A008
;  1674          1782  5              vms_desc_source[dsc$w_length]  = .Length;                           ! A008
;  1675          1783  5              vms_desc_source[dsc$a_pointer] = .Buffer;                           ! A008
;  1676          1784  5
;  1677          1785  5              !++
;  1678          1786  5              ! Set up the target
;  1679          1787  5              !--
;  1680          1788  5              vms_desc_target[dsc$b_class]   =  dsc$k_class_s;                    !
;  1681          1789  5              vms_desc_target[dsc$b_dtype]   = .Data_Type[.context];              !
;  1682          1790  5              vms_desc_target[dsc$w_length]  = .Data_Size[.context];              !
;  1683          1791  5              vms_desc_target[dsc$a_pointer] =  Op_Buffer;                        !
;  1684          1792  5
;  1685          1793  5              Op_Buffer[0,0,32,0] = 0;     ! Clear high operand bytes
;  1686          1794  5              Op_Buffer[4,0,32,0] = 0;     ! to extend to octaword value
;  1687          1795  5              Op_Buffer[8,0,32,0] = 0;     !
;  1688          1796  5              Op_Buffer[12,0,32,0] = 0;    !
;  1689          1797  5
;  1690          1798  5              Val_desc_source = dbg$make_val_desc( vms_desc_source, dbg$k_v_value_desc );! Convert to a value
;  1691          1799  5
;  1692          1800  5              !++
;  1693          1801  5              ! Test for negatives and fix things up
;  1694          1802  5              !
;  1695          1803  5              ! DBGCONV_TEXT_VALUE does not accept the negative sign
;  1696          1804  5              ! in the value, but it requires a flag in the value
;  1697          1805  5              ! descriptor.
```

```
;1698    1806  5          !--
;1699    1807  5          IF CH$RCHAR( .Val_desc_source[ DBG$L_VALUE_POINTER ] ) EQL %C'-'
;1700    1808  5          THEN
;1701    1809  6              BEGIN
;1702    1810  6              Val_desc_source[ DBG$L_VALUE_POINTER ] = .Val_desc_source[ DBG$L_VALUE_POINTER ] + 1;
;1703    1811  6              Val_desc_source[ DBG$W_VALUE_LENGTH  ] = .Val_desc_source[ DBG$W_VALUE_LENGTH  ] - 1;
;1704    1812  6              Val_desc_source[ DBG$W_VALUE_SIGN_CODE ] = TOKEN$K_NEGCONST;
;1705    1813  5              END;
;1706    1814  5
;1707    1815  5          !++
;1708    1816  5          ! Send in the right radix so it will do the right conversion
;1709    1817  5          !--
;1710    1818  5          Val_desc_source[ DBG$W_VALUE_TOKENCODE ] =
;1711    1819  6              (SELECTONE .DBG$GB_RADIX[DBG$B_RADIX_INPUT] OF
;1712    1820  6                  SET
;1713    1821  6                  [DBG$K_BINARY]  : TOKEN$K_BIN_INTEGER;
;1714    1822  6                  [DBG$K_OCTAL]   : TOKEN$K_OCT_INTEGER;
;1715    1823  6                  [DBG$K_HEX]     : TOKEN$K_HEX_INTEGER;
;1716    1824  6                  [DBG$K_DECIMAL] : TOKEN$K_INTEGER;
;1717    1825  6                  [OTHERWISE]     : $DBG_ERROR( 'DBGENCDEC\PARSER_OPERAND unexpected radix');
;1718    1826  5                  TES);
;1719    1827  5
;1720    1828  5          !++
;1721    1829  5          ! Make the target value descriptor
;1722    1830  5          !--
;1723    1831  5          Val_desc_target = dbg$make_val_desc( vms_desc_target, dbg$k_value_desc );
;1724    1832  5
;1725    1833  5          Val_desc_target = DBG$CONV_TEXT_VALUE( .Val_desc_source,                    ! Convert the number
;1726    1834  5                                                 .Val_desc_target,                    !
;1727    1835  5                                                 .vms_desc_target[dsc$b_dtype]);      !
;1728    1836  5
;1729    1837  5          !++
;1730    1838  5          ! Move the converted value in
;1731    1839  5          ! normally it is already in  Op_buffer but in the case
;1732    1840  5          ! of negative numbers the routines used change the
;1733    1841  5          ! pointer rather than move the data.
;1734    1842  5          !--
;1735    1843  5          CH$MOVE( .Val_desc_target[ DBG$W_VALUE_LENGTH ],
;1736    1844  5                   CH$PTR( .Val_desc_target[ DBG$L_VALUE_POINTER ] ),
;1737    1845  5                   CH$PTR( Op_buffer ) );
;1738    1846  5
;1739    1847  6          IF (.R_Mode<16,8,0> EQLU %C'S') OR  ((.R_Mode<16,8,0> EQL 0) AND
;1740    1848  6              (.Op_Buffer[0,0,32,0] LSSU %X'40') AND (.Op_Buffer[4,0,32,0] EQL 0) AND
;1741    1849  7              ((.Context EQL context_b) OR (.Context EQL context_bu) OR
;1742    1850  7               (.Context EQL context_w) OR (.Context EQL context_wu) OR
;1743    1851  5               (.Context EQL context_l) OR (.Context EQL context_q))) THEN
;1744    1852  6              BEGIN
;1745    1853  6              BUILTIN ROT;
;1746    1854  6              IF (.Context EQL context_f) THEN
;1747    1855  6                  Operand_value = ROT(T.Operand_Value XOR %X'4000'),-4);
;1748    1856  6              IF (.Operand_Value GTRU %X'3F') THEN Report_Error;
;1749    1857  6              Store_Operand(.EnCode,Operand_Value,1);
;1750    1858  6              END
;1751    1859  5          ELSE
;1752    1860  6              BEGIN
;1753    1861  6              Store_Operand(.EnCode,UPLIT BYTE(%X'8F'),1);
;1754    1862  6              Store_Operand(.EnCode,Operand_Value,.Data_Size[.Context]);
```

```
; 1755          1863  5                END;
; 1756          1864  5            LEAVE scan;
; 1757          1865  4            END;
; 1758          1866  3        END;                                    ! End Addr.
```

```
;1760    1867  3          !+
;1761    1868  3          ! Here if we have an register/displacement operand,
;1762    1869  3          ! or an absolute address (explicit or implicit).
;1763    1870  3          !
;1764    1871  3          IF NOT Parse_Expression(-1,.Buffer,.Length,Address) THEN SIGNAL(DBG$_INVEXPR, 1, ..Operand_number);
;1765    1872  3          IF (Check_Register(.Address) GEQ 0) THEN SIGNAL(DBG$_INVEXPR, 1, ..Operand_number);
;1766    1873  3
;1767    1874  3          IF (.R_Mode EQL %X'100') THEN R_Mode =                    ! G^<address>
;1768    1875  4              (IF (.Address GEQU %X'40000000') THEN %X'90' ELSE %X'E0');
;1769    1876  3
;1770    1877  3          IF .Displacement_size_needed THEN                                    !A005 We need to chose a disp
;1771    1878  5               R_Mode = (IF (.Regnum EQL 15)                                   !A005 PC relative displacemen
;1772    1879  4                         THEN                                                  !A005 Yes
;1773    1880  5                           (SELECTONE .Address -                               !A005 Calculate the offset wi
;1774    1881  6                                      (1 + .Encode[Enc_Final_Address] +        !A005 the operand since we do
;1775    1882  5                                      .EnCode[Enc_Output_Length]) OF           !A005
;1776    1883  5                           SET                                                 !A005
;1777    1884  5                           [-127     TO 128]:      %X'A0';                   !A005 Byte  Note that the tab
;1778    1885  5                           [-32766   TO 32769]:    %X'C0';                   !A005 Word  take the lack of
;1779    1886  5                           [OTHERWISE]:            %X'E0';                   !A005 Long  calculation into
;1780    1887  5                           TES)                                              !A005
;1781    1888  4                         ELSE
;1782    1889  5                           (SELECTONE .Address OF                            !A005
;1783    1890  5                           SET                                               !A005
;1784    1891  5                           [-128     TO 127]:              %X'A0';         !A005 Byte
;1785    1892  5                           [-32768   TO 32767]:            %X'C0';         !A005 Word
;1786    1893  5                           [OTHERWISE]:                    %X'E0';         !A005 Long
;1787    1894  5                           TES));                                            !A005
;1788    1895  3
;1789    1896  5          Length = (IF ((.R_Mode EQL %X'90') OR (.R_Mode EQL %X'80'))       ! Long operand?
;1790    1897  4                    THEN 4
;1791    1898  3                    ELSE 1^((.R_Mode-%X'A0')/%X'20'));
;1792    1899  4          IF (.Regnum EQL 15) AND (.R_Mode NEQ %X'80') AND (.R_Mode NEQ %X'90')  ! Regnum of 15 means PC relat
;1793    1900  4          THEN Address = .Address - (1 + .Length + .Encode[Enc_Final_Address] +  ! Absolute and immediate
;1794    1901  3                                                  .EnCode[Enc_Output_Length]);
;1795    1902  3
;1796    1903  3          R_Mode = .R_Mode + .Defer + .Regnum;
;1797    1904  3          Store_Operand(.EnCode,R_Mode,1);
;1798    1905  3          Store_Operand(.EnCode,Address,.Length);
;1799    1906  2          END;                          ! End of block 'scan'
;1800    1907  2
;1801    1908  2      EnCode[Enc_Input_Buffer] = .EnCode[Enc_Input_Buffer] + .Parsed;
;1802    1909  2      EnCode[Enc_Input_Length] = .EnCode[Enc_Input_Length] - .Parsed;
;1803    1910  2      IF .EnCode[Enc_Input_Length] GTR 0 THEN
;1804    1911  3          BEGIN
;1805    1912  3          EnCode[Enc_Input_Buffer] = .EnCode[Enc_Input_Buffer] + 1;
;1806    1913  3          EnCode[Enc_Input_Length] = .EnCode[Enc_Input_Length] - 1;
;1807    1914  2          END;
;1808    1915  1      END;


                                                        .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

                                    53  49  47  4C  57  42  0119F P.AAT:  .ASCII  \BWLGIS\
53  52  41  50  5C  43  45  44  43  4E  45  47  42  44  29  011A5 P.AAU:  .ASCII  \)DBGENCDEC\<92>\PARSER_OPERAND unexpect\
78  65  6E  75  20  44  4E  41  52  45  50  4F  5F  52  45  011B4
```

```
                         74 63 65 70  011C3                                               ;
             78 69 64 61 72 20 64 65  011C7          .ASCII  \ed_radix\                   ;
                                  8F  011CF P.AAV:   .BYTE   -113                         ;

                                        MODE_CHAR=           P.AAT

                                                     .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                              0FFC 00000 PARSE_OPERAND:
                                                     .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11  ; 1516
                     5E           20  C2 00002       SUBL2   #32, SP                      ;
                     55     08    AC  D0 00005       MOVL    CONTEXT, R5                  ; 1530
                     0B           55  D1 00009       CMPL    R5, #11                      ; 1532
                                  13  19 0000C       BLSS    1$                           ;
                     0C           55  D1 0000E       CMPL    R5, #12                      ;
                                  0E  14 00011       BGTR    1$                           ;
                            0C    AC  DD 00013       PUSHL   OPERAND_NUMBER               ; 1534
                                  0A  DD 00016       PUSHL   #10                          ;
                            04    AC  DD 00018       PUSHL   ENCODE                       ;
                 E1  AF           03  FB 0001B       CALLS   #3, PARSE_OPERAND            ;
                                  28  11 0001F       BRB     2$                           ; 1535
                     0D           55  D1 00021 1$:   CMPL    R5, #13                      ; 1538
                                  30  19 00024       BLSS    3$                           ;
                     0E           55  D1 00026       CMPL    R5, #14                      ;
                                  2B  14 00029       BGTR    3$                           ;
                            0C    AC  DD 0002B       PUSHL   OPERAND_NUMBER               ; 1540
                                  02  DD 0002E       PUSHL   #2                           ;
                            04    AC  DD 00030       PUSHL   ENCODE                       ;
                 C9  AF           03  FB 00033       CALLS   #3, PARSE_OPERAND            ;
                            0C    AC  DD 00037       PUSHL   OPERAND_NUMBER               ; 1541
                                  7E  D4 0003A       CLRL    -(SP)                        ;
                            04    AC  DD 0003C       PUSHL   ENCODE                       ;
                 BD  AF           03  FB 0003F       CALLS   #3, PARSE_OPERAND            ;
                     0D           55  D1 00043       CMPL    R5, #13                      ; 1542
                                  01  13 00046       BEQL    2$                           ;
                                      04 00048       RET                                  ;
                            0C    AC  DD 00049 2$:   PUSHL   OPERAND_NUMBER               ;
                                  7E  D4 0004C       CLRL    -(SP)                        ;
                            04    AC  DD 0004E       PUSHL   ENCODE                       ;
                 AB  AF           03  FB 00051       CALLS   #3, PARSE_OPERAND            ;
                                      04 00055       RET                                  ; 1539
                            0C    BC  D6 00056 3$:   INCL    @OPERAND_NUMBER              ; 1548
                                  2C  DD 00059       PUSHL   #44                          ; 1549
                     52     04    AC  D0 0005B       MOVL    ENCODE, R2                   ;
                                  52  DD 0005F       PUSHL   R2                           ;
              0000V  CF           02  FB 00061       CALLS   #2, SCAN_OPERAND             ;
                     6E           50  D0 00066       MOVL    R0, PARSED                   ;
                     56           6E  D0 00069       MOVL    PARSED, LENGTH               ;
                     57     04    A2  D0 0006C       MOVL    4(R2), BUFFER                ; 1550
                                  56  D5 00070 4$:   TSTL    LENGTH                       ; 1551
                                  0B  15 00072       BLEQ    5$                           ;
                     20     FF  A647  91 00074       CMPB    -1(LENGTH)[BUFFER], #32      ;
                                  04  12 00079       BNEQ    5$                           ;
                                  56  D7 0007B       DECL    LENGTH                       ;
                                  F1  11 0007D       BRB     4$                           ;
                                  56  D5 0007F 5$:   TSTL    LENGTH                       ; 1552
```

```
                                 12  14 00081          BGTR    6$                                      ;
                              0C BC  DD 00083          PUSHL   @OPERAND_NUMBER                         ;
                                 01  DD 00086          PUSHL   #1                                      ;
                          00028268 8F  DD 00088          PUSHL   #164456                                 ;
                00000000G   00     03  FB 0008E          CALLS   #3, LIB$SIGNAL                          ;
                              08 AE  D4 00095 6$:        CLRL    R_MODE                                  ; 1563
                           58        01  CE 00098          MNEGL   #1, INDEXED_REGNUM                      ; 1564
                                     5A  D4 0009B          CLRL    DISPLACEMENT_SIZE_NEEDED                ; 1565
                        5D 8F    FF A647  91 0009D          CMPB    -1(LENGTH)[BUFFER], #93                 ; 1566
                                    03  13 000A3          BEQL    7$                                      ;
                                  008E  31 000A5          BRW     14$                                     ;
                           52           56  D0 000A8 7$:     MOVL    LENGTH, START                           ; 1568
                                  0080  31 000AB          BRW     12$                                     ;
                        5B 8F    FF A247  91 000AE 8$:      CMPB    -1(START)[BUFFER], #91                  ; 1569
                                    78  12 000B4          BNEQ    12$                                     ;
                50              56       52  C3 000B6          SUBL3   START, LENGTH, R0                       ; 1571
                                 FF  A0  9F 000BA          PUSHAB  -1(R0)                                  ;
                                   6247  9F 000BD          PUSHAB  (START)[BUFFER]                         ;
                          0000V CF       02  FB 000C0          CALLS   #2, PARSE_REGISTER                      ;
                            04 AE        50  D0 000C5          MOVL    R0, REGNUM                              ;
                           53        04 AE  D0 000C9          MOVL    REGNUM, R3                              ; 1572
                                    12  18 000CD          BGEQ    9$                                      ;
                              0C BC  DD 000CF          PUSHL   @OPERAND_NUMBER                         ; 1574
                                 01  DD 000D2          PUSHL   #1                                      ;
                          00028278 8F  DD 000D4          PUSHL   #164472                                 ;
                00000000G   00     03  FB 000DA          CALLS   #3, LIB$SIGNAL                          ;
                           0F        53  D1 000E1 9$:     CMPL    R3, #15                                 ; 1575
                                    12  19 000E4          BLSS    10$                                     ;
                              0C BC  DD 000E6          PUSHL   @OPERAND_NUMBER                         ; 1577
                                 01  DD 000E9          PUSHL   #1                                      ;
                          00028270 8F  DD 000EB          PUSHL   #164464                                 ;
                00000000G   00     03  FB 000F1          CALLS   #3, LIB$SIGNAL                          ;
                           56    FF  A2  9E 000F8 10$:     MOVAB   -1(R2), LENGTH                          ; 1578
                                    12  14 000FC          BGTR    11$                                     ;
                              0C BC  DD 000FE          PUSHL   @OPERAND_NUMBER                         ; 1580
                                 01  DD 00101          PUSHL   #1                                      ;
                          00028268 8F  DD 00103          PUSHL   #164456                                 ;
                00000000G   00     03  FB 00109          CALLS   #3, LIB$SIGNAL                          ;
                           58        53  D0 00110 11$:     MOVL    R3, INDEXED_REGNUM                      ; 1581
                        04 AE 00000040 8F  C0 00113          ADDL2   #64, REGNUM                             ; 1582
                                 01  DD 0011B          PUSHL   #1                                      ; 1583
                              08 AE  9F 0011D          PUSHAB  REGNUM                                  ;
                              04 AC  DD 00120          PUSHL   ENCODE                                  ;
                          0000V CF       03  FB 00123          CALLS   #3, STORE_OPERAND                       ;
                            08 AE        01  CE 00128          MNEGL   #1, R_MODE                              ; 1584
                                    08  11 0012C          BRB     14$                                     ; 1570
                           02           52  F5 0012E 12$:     SOBGTR  START, 13$                              ; 1569
                                    03  11 00131          BRB     14$                                     ;
                                  FF78  31 00133 13$:      BRW     8$                                      ;
                        2B 8F    FF A647  91 00136 14$:     CMPB    -1(LENGTH)[BUFFER], #43                 ; 1589
                                    1A  12 00138          BNEQ    15$                                     ;
                     08 AE 80       8F  9A 0013D          MOVZBL  #128, R_MODE                            ; 1591
                        12          56  F5 00142          SOBGTR  LENGTH, 15$                             ; 1592
                              0C BC  DD 00145          PUSHL   @OPERAND_NUMBER                         ; 1594
                                 01  DD 00148          PUSHL   #1                                      ;
                          00028268 8F  DD 0014A          PUSHL   #164456                                 ;
                00000000G   00     03  FB 00150          CALLS   #3, LIB$SIGNAL                          ;
```

```
                                29        FF A647   91 00157 15$:     CMPB    -1(LENGTH)[BUFFER], #41                 ; 1597
                                                03   13 0015C         BEQL    16$
                                             00F0   31 0015E         BRW     30$
                                52                56   D0 00161 16$:     MOVL    LENGTH, START                       ; 1599
                                             00E2   31 00164 17$:     BRW     28$
                                28        FF A247   91 00167 18$:     CMPB    -1(START)[BUFFER], #40                  ; 1600
                                                F6   12 0016C         BNEQ    17$
                     50         56                52   C3 0016E         SUBL3   START, LENGTH, R0                   ; 1602
                                             FF A0   9F 00172         PUSHAB  -1(R0)
                                              6247   9F 00175         PUSHAB  (START)[BUFFER]
                           0000V CF                02   FB 00178         CALLS   #2, PARSE_REGISTER
                              04 AE                50   D0 0017D         MOVL    R0, REGNUM
                                53        04      AE   D0 00181         MOVL    REGNUM, R3                          ; 1603
                                0F                53   D1 00185         CMPL    R3, #15
                                                  12   12 00188         BNEQ    19$
                                             0C   BC   DD 0018A         PUSHL   @OPERAND_NUMBER                     ; 1605
                                             01   DD 0018D             PUSHL   #1
                                       00028270   8F   DD 0018F         PUSHL   #164464
                  00000000G    00                03   FB 00195         CALLS   #3, LIB$SIGNAL
                                             08   AE   D5 0019C 19$:     TSTL    R_MODE                              ; 1606
                                                  3D   15 0019F         BLEQ    22$
                                                  53   D5 001A1         TSTL    R3                                  ; 1608
                                                  12   18 001A3         BGEQ    20$
                                             0C   BC   DD 001A5         PUSHL   @OPERAND_NUMBER                     ; 1610
                                             01   DD 001A8             PUSHL   #1
                                       00028278   8F   DD 001AA         PUSHL   #164472
                  00000000G    00                03   FB 001B0         CALLS   #3, LIB$SIGNAL
                                53                58   D1 001B7 20$:     CMPL    INDEXED_REGNUM, R3                  ; 1611
                                                  1C   12 001BA         BNEQ    21$
                  00000080     8F        08      AE   D1 001BC         CMPL    R_MODE, #128                        ; 1612
                                                  12   12 001C4         BNEQ    2T$
                                             0C   BC   DD 001C6         PUSHL   @OPERAND_NUMBER                     ; 1614
                                             01   DD 001C9             PUSHL   #1
                                       00028280   8F   DD 001CB         PUSHL   #164480
                  00000000G    00                03   FB 001D1         CALLS   #3, LIB$SIGNAL
                                56        FF      A2   9E 001D8 21$:     MOVAB   -1(R2), LENGTH                      ; 1615
                                                  78   11 001DC         BRB     31$                                 ; 1616
                                                  53   D5 001DE 22$:     TSTL    R3                                  ; 1618
                                                  12   18 001E0         BGEQ    23$
                                             0C   BC   DD 001E2         PUSHL   @OPERAND_NUMBER                     ; 1620
                                             01   DD 001E5             PUSHL   #1
                                       00028278   8F   DD 001E7         PUSHL   #164472
                  00000000G    00                03   FB 001ED         CALLS   #3, LIB$SIGNAL
                                01                52   D1 001F4 23$:     CMPL    START, #1                           ; 1621
                                                  07   12 001F7         BNEQ    24$
                           08   AE        60      8F   9A 001F9         MOVZBL  #96, R_MODE                         ; 1622
                                                  0F   11 001FE         BRB     25$
                                2D                67   91 00200 24$:     CMPB    (BUFFER), #45                       ; 1623
                                                  0A   12 00203         BNEQ    25$
                                02                52   D1 00205         CMPL    START, #2
                                                  05   12 00208         BNEQ    25$
                           08   AE        70      8F   9A 0020A         MOVZBL  #112, R_MODE                        ; 1624
                                54        08      AE   D0 0020F 25$:     MOVL    R_MODE, R4                          ; 1625
                                                  26   15 00213         BLEQ    27$
                                53                58   D1 00215         CMPL    INDEXED_REGNUM, R3                  ; 1627
                                                  1B   12 00218         BNEQ    26$
                  00000070     8F                54   D1 0021A         CMPL    R4, #112                            ; 1628
```

```
                                    12  12 00221          BNEQ    26$                                     ;
                                 0C BC  DD 00223          PUSHL   @OPERAND_NUMBER                         ; 1630
                                    01  DD 00226          PUSHL   #1                                      ;
                              00028280  8F  DD 00228          PUSHL   #164480                             ;
                 00000000G   00         03  FB 00022E          CALLS   #3, LIB$SIGNAL                  ;
                          04 AE         54  C0 00235 26$:      ADDL2   R4, REGNUM                      ; 1631
                                    5D  11 00239          BRB     35$                                     ; 1632
                             56 FF      A2  9E 0023B 27$:      MOVAB   -1(R2), LENGTH                  ; 1635
                          08 AE      A0  8F  9A 0023F          MOVZBL  #160, R_MODE                    ; 1636
                             5A         01  D0 00244          MOVL    #1, DISPLACEMENT_SIZE_NEEDED    ; 1637
                                    0D  11 00247          BRB     31$                                     ; 1638
                             02         52  F5 00249 28$:      SOBGTR  START, 29$                      ; 1600
                                    08  11 0024C          BRB     31$                                     ; 1597
                                FF16    31  0024E 29$:      BRW     18$                                     ; 1600
                                 08 AE  D5  00251 30$:      TSTL    R_MODE                          ; 1641
                                    65  14 00254          BGTR    38$                                     ;
                                    56  D5 00256 31$:      TSTL    LENGTH                          ; 1655
                                    0F  15 00258          BLEQ    32$                                     ;
                          40 8F         67  91 0025A          CMPB    (BUFFER), #64                   ;
                                    09  12 0025E          BNEQ    32$                                     ;
                                    56  D7 00260          DECL    LENGTH                          ; 1657
                                    57  D6 00262          INCL    BUFFER                          ; 1658
                             59         10  D0 00264          MOVL    #16, DEFER                      ; 1659
                                    02  11 00267          BRB     33$                                     ; 1655
                                    59  D4 00269 32$:      CLRL    DEFER                           ; 1662
                             52 08 AE   D0  0026B 33$:      MOVL    R_MODE, R2                      ; 1664
                 00000080 8F            52  D1 0026F          CMPL    R2, #128                        ;
                                    22  12 00276          BNEQ    36$                                     ;
                                    56  D5 00278          TSTL    LENGTH                          ; 1666
                                    12  13 0027A          BEQL    34$                                     ;
                                 0C BC  DD 0027C          PUSHL   @OPERAND_NUMBER                         ;
                                    01  DD 0027F          PUSHL   #1                                      ;
                              00028210  8F  DD 00281          PUSHL   #164368                             ;
                    00000000G   00         03  FB 00287          CALLS   #3, LIB$SIGNAL                  ;
                50                  52 04 AE   C1  0028E 34$:      ADDL3   REGNUM, R2, R0                  ; 1667
          04 AE                     50         59  C1 00293          ADDL3   DEFER, R0, REGNUM               ;
                                    7E  11 00298 35$:      BRB     46$                                     ; 1668
                                    56  D5 0029A 36$:      TSTL    LENGTH                          ; 1675
                                    12  14 0029C          BGTR    37$                                     ;
                                 0C BC  DD 0029E          PUSHL   @OPERAND_NUMBER                         ;
                                    01  DD 002A1          PUSHL   #1                                      ;
                              00028268  8F  DD 002A3          PUSHL   #164456                             ;
                    00000000G   00         03  FB 002A9          CALLS   #3, LIB$SIGNAL                  ;
                                23         67  91 002B0 37$:      CMPB    (BUFFER), #35                   ; 1678
                                    21  12 002B3          BNEQ    41$                                     ;
                                    57  D6 002B5          INCL    BUFFER                          ; 1680
                                    56  D7 002B7          DECL    LENGTH                          ; 1681
                                    03  14 002B9          BGTR    39$                                     ; 1682
                                025A    31  002BB 38$:      BRW     69$                                     ;
                                    52  D5 002BE 39$:      TSTL    R2                                      ;
                                    F9  14 002C0          BGTR    38$                                     ;
                                    59  D5 002C2          TSTL    DEFER                           ; 1683
                                    04  12 002C4          BNEQ    40$                                     ;
                                    52  D5 002C6          TSTL    R2                                      ;
                                    F1  19 002C8          BLSS    38$                                     ;
                          08 AE      80  8F  9A 002CA 40$:      MOVZBL  #128, R_MODE                    ; 1684
                          04 AE          0F  D0 002CF          MOVL    #15, REGNUM                     ; 1685
```

```
                                00EB  31 002D3          BRW     57$                                     ; 1678
                          02      56  D1 002D6 41$:     CMPL    LENGTH, #2                              ; 1689
                                  07  19 002D9          BLSS    42$                                     ;
                       5E 8F  01  A7  91 002DB          CMPB    1(BUFFER), #94                          ;
                                  3E  13 002E0          BEQL    47$                                     ;
                                  52  D5 002E2 42$:     TSTL    R2                                      ; 1691
                                  11  12 002E4          BNEQ    43$                                     ;
                                  59  D5 002E6          TSTL    DEFER                                   ;
                                  0D  12 002E8          BNEQ    43$                                     ;
                                  56  DD 002EA          PUSHL   LENGTH                                  ; 1692
                                  57  DD 002EC          PUSHL   BUFFER                                  ;
                    0000V CF      02  FB 002EE          CALLS   #2, PARSE_REGISTER                      ;
                                  50  D5 002F3          TSTL    R0                                      ;
                                  13  18 002F5          BGEQ    45$                                     ;
                                  52  D5 002F7 43$:     TSTL    R2                                      ; 1695
                                  0C  14 002F9          BGTR    44$                                     ;
                       04 AE      0F  D0 002FB          MOVL    #15, REGNUM                             ; 1698
                       08 AE   A0 8F  9A 002FF          MOVZBL  #160, R_MODE                            ; 1699
                          5A      01  D0 00304          MOVL    #1, DISPLACEMENT_SIZE_NEEDED            ; 1700
                                0246  31 00307 44$:     BRW     73$                                     ; 1702
                                  56  DD 0030A 45$:     PUSHL   LENGTH                                  ; 1704
                                  57  DD 0030C          PUSHL   BUFFER                                  ;
                    0000V CF      02  FB 0030E          CALLS   #2, PARSE_REGISTER                      ;
                       04 AE   50 A0  9E 00313          MOVAB   80(R0), REGNUM                          ;
                                  01  DD 00318 46$:     PUSHL   #1                                      ; 1705
                                  08 AE 9F 0031A        PUSHAB  REGNUM                                  ;
                                037E  31 0031D          BRW     91$                                     ;
                                  52  D4 00320 47$:     CLRL    INDEX                                   ; 1711
          00000000'EF 42          67  91 00322 48$:     CMPB    (BUFFER), MODE_CHAR[INDEX]              ; 1712
                                  7D  12 0032A          BNEQ    55$                                     ;
                          57      02  C0 0032C          ADDL2   #2, BUFFER                              ; 1714
                          56      02  C2 0032F          SUBL2   #2, LENGTH                              ; 1715
                                  12  14 00332          BGTR    49$                                     ; 1716
                             0C   BC  DD 00334          PUSHL   @OPERAND_NUMBER                         ; 1718
                                  01  DD 00337          PUSHL   #1                                      ;
                       00028268   8F  DD 00339          PUSHL   #164456                                 ;
             00000000G 00         03  FB 0033F          CALLS   #3, LIB$SIGNAL                          ;
                          03      52  D1 00346 49$:     CMPL    INDEX, #3                               ; 1719
                                  25  14 00349          BGTR    52$                                     ;
                                  09  12 0034B          BNEQ    50$                                     ; 1721
                             08   AE  D5 0034D          TSTL    R_MODE                                  ; 1722
                                  2A  14 00350          BGTR    53$                                     ;
                                  59  D5 00352          TSTL    DEFER                                   ;
                                  26  12 00354          BNEQ    53$                                     ;
                             08   AE  D5 00356 50$:     TSTL    R_MODE                                  ; 1724
                                  09  14 00359          BGTR    51$                                     ;
                       08 AE      A0  8F 9A 0035B          MOVZBL  #160, R_MODE                        ; 1726
                       04 AE      0F  D0 00360          MOVL    #15, REGNUM                             ; 1727
                50        52      05  78 00364 51$:     ASHL    #5, INDEX, R0                           ; 1729
                       08 AE      50  C0 00368          ADDL2   R0, R_MODE                              ;
                                  5A  D4 0036C          CLRL    DISPLACEMENT_SIZE_NEEDED                ; 1730
                                  51  11 0036E          BRB     57$                                     ; 1719
                             08   AE  D5 00370 52$:     TSTL    R_MODE                                  ; 1734
                                  07  14 00373          BGTR    53$                                     ;
                                  59  D5 00375          TSTL    DEFER                                   ;
                                  03  14 00377          BGTR    53$                                     ;
                          23      67  91 00379          CMPB    (BUFFER), #35                           ; 1735
```

```
                                   03   13 0037C 53$:     BEQL    54$                              ;
                                 0197   31 0037E          BRW     69$                              ;
                   50 00000000'EF42   9A 00381 54$:     MOVZBL  MODE_CHAR[INDEX], R0             ; 1736
             50          50      10   78 00389          ASHL    #16, R0, R0                      ;
             08          AE 0080 C0   9E 0038D          MOVAB   128(R0), R_MODE                  ;
             04          AE      0F   D0 00393          MOVL    #15, REGNUM                      ; 1737
                                 57   D6 00397          INCL    BUFFER                           ; 1738
                         25      56   F5 00399          SOBGTR  LENGTH, 57$                      ; 1739
                            0C   BC   DD 0039C          PUSHL   @OPERAND_NUMBER                  ; 1742
                                 01   DD 0039F          PUSHL   #1                               ;
                      00028268   8F   DD 003A1          PUSHL   #164456                          ;
                                 11   11 003A7          BRB     56$                              ;
  FF73       52          01      05   F1 003A9 55$:     ACBL    #5, #1, INDEX, 48$               ; 1712
                            0C   BC   DD 003AF          PUSHL   @OPERAND_NUMBER                  ; 1746
                                 01   DD 003B2          PUSHL   #1                               ;
                      00028288   8F   DD 003B4          PUSHL   #164488                          ;
          00000000G      00      03   FB 003BA 56$:     CALLS   #3, LIB$SIGNAL                   ;
       50        08      AE 04   AE   C1 003C1 57$:     ADDL3   REGNUM, R_MODE, R0               ; 1749
                         50      59   C0 003C7          ADDL2   DEFER, R0                        ;
                  8F     8F      50   91 003CA          CMPB    R0, #143                         ;
                                 03   13 003CE          BEQL    58$                              ;
                                017D  31 003D0          BRW     73$                              ;
                                 5B   D4 003D3 58$:     CLRL    R11                              ; 1767
                  53     8F 0A   AE   91 003D5          CMPB    R_MODE+2, #83                    ;
                                 16   12 003DA          BNEQ    60$                              ;
                                 5B   D6 003DC          INCL    R11                              ;
                         05      55   D1 003DE          CMPL    R5, #5                           ; 1770
                                 0B   19 003E1          BLSS    59$                              ;
                         08      55   D1 003E3          CMPL    R5, #8                           ;
                                 06   14 003E6          BGTR    59$                              ;
                  08     AC      05   D0 003E8          MOVL    #5, CONTEXT                      ; 1772
                                 04   11 003EC          BRB     60$                              ;
                  08     AC      02   D0 003EE 59$:     MOVL    #2, CONTEXT                      ; 1774
                  1A     AE 010E 8F   B0 003F2 60$:     MOVW    #270, VMS_DESC_SOURCE+2          ; 1781
                  18     AE      56   B0 003F8          MOVW    LENGTH, VMS_DESC_SOURCE          ; 1782
                  1C     AE      57   D0 003FC          MOVL    BUFFER, VMS_DESC_SOURCE+4        ; 1783
                  13     AE      01   90 00400          MOVB    #1, VMS_DESC_TARGET+3            ; 1788
                         58 08   AC   D0 00404          MOVL    CONTEXT, R8                      ; 1789
                  12     AE 00000000'EF48 90 00408      MOVB    DATA_TYPE[R8], VMS_DESC_TARGET+2 ;
                  10     AE 00000000'EF48 9B 00411      MOVZBW  DATA_SIZE[R8], VMS_DESC_TARGET   ; 1790
                  14     AE 00000000'  EF 9E 0041A      MOVAB   OP_BUFFER, VMS_DESC_TARGET+4     ; 1791
                            00000000'  EF 7C 00422      CLRQ    OP_BUFFER                        ; 1793
                            00000000'  EF 7C 00428      CLRQ    OP_BUFFER+8                      ; 1795
                         7E      83   8F 9A 0042E       MOVZBL  #131, -(SP)                      ; 1798
                                 1C   AE 9F 00432       PUSHAB  VMS_DESC_SOURCE                  ;
          00000000G      00      02   FB 00435          CALLS   #2, DBG$MAKE_VAL_DESC            ;
                         52      50   D0 0043C          MOVL    R0, VAL_DESC_SOURCE              ;
                         2D 18   B2   91 0043F          CMPB    @24(VAL_DESC_SOURCE), #45        ; 1807
                                 0B   12 00443          BNEQ    61$                              ;
                            18   A2   D6 00445          INCL    24(VAL_DESC_SOURCE)              ; 1810
                            14   A2   B7 00448          DECW    20(VAL_DESC_SOURCE)              ; 1811
                  12     A2 42   8F   9B 0044B          MOVZBW  #66, 18(VAL_DESC_SOURCE)         ; 1812
                         50 00000000G 00 9A 00450 61$:  MOVZBL  DBG$GB_RADIX, R0                 ; 1819
                         02      50   91 00457          CMPB    R0, #2                           ; 1821
                                 05   12 0045A          BNEQ    62$                              ;
                         50      0A   D0 0045C          MOVL    #10, R0                          ;
                                 33   11 0045F          BRB     66$                              ;
```

```
                              08            50  91 00461 62$:   CMPB    R0, #8                                  ; 1822
                                            05  12 00464        BNEQ    63$                                     ;
                              50            0B  D0 00466        MOVL    #11, R0                                 ;
                                            29  11 00469        BRB     66$                                     ;
                              10            50  91 0046B 63$:   CMPB    R0, #16                                 ; 1823
                                            05  12 0046E        BNEQ    64$                                     ;
                              50            05  D0 00470        MOVL    #5, R0                                  ;
                                            1F  11 00473        BRB     66$                                     ;
                              0A            50  91 00475 64$:   CMPB    R0, #10                                 ; 1824
                                            05  12 00478        BNEQ    65$                                     ;
                              50            04  D0 0047A        MOVL    #4, R0                                  ;
                                            15  11 0047D        BRB     66$                                     ;
                                 00000000'  EF  9F 0047F 65$:   PUSHAB  P.AAU                                   ; 1825
                                            01  DD 00485        PUSHL   #1                                      ;
                                 00028362   8F  DD 00487        PUSHL   #164706                                 ;
                   00000000G  00            03  FB 0048D        CALLS   #3, LIB$SIGNAL                          ;
                          10  A2            50  B0 00494 66$:   MOVW    R0, 16(VAL_DESC_SOURCE)                 ; 1819
                              7E        7A  8F  9A 00498        MOVZBL  #122, -(SP)                             ; 1831
                                        14  AE  9F 0049C        PUSHAB  VMS_DESC_TARGET                         ;
                   00000000G  00            02  FB 0049F        CALLS   #2, DBG$MAKE_VAL_DESC                   ;
                              7E        12  AE  9A 004A6        MOVZBL  VMS_DESC_TARGET+2, -(SP)                ; 1835
                                            50  DD 004AA        PUSHL   VAL_DESC_TARGET                         ; 1834
                                            52  DD 004AC        PUSHL   VAL_DESC_SOURCE                         ; 1833
                   00000000G  00            03  FB 004AE        CALLS   #3, DBG$CONV_TEXT_VALUE                 ;
      00000000'  EF           18  B0    14  A0  28 004B5        MOVC3   20(VAL_DESC_TARGET), @24(VAL_DESC_TARGET), -; 1845
                                                                        OP_BUFFER                               ;
                              33            5B  E8 004BF        BLBS    R11, 67$                                ; 1857
                                        0A  AE  95 004C2        TSTB    R_MODE+2                                ; 1847
                                            68  12 004C5        BNEQ    71$                                     ;
                              3F 00000000'  EF  D1 004C7        CMPL    OP_BUFFER, #63                          ; 1848
                                            5F  1A 004CE        BGTRU   71$                                     ;
                                 00000000'  EF  D5 004D0        TSTL    OP_BUFFER+4                             ;
                                            57  12 004D6        BNEQ    71$                                     ;
                                            58  D5 004D8        TSTL    R8                                      ; 1849
                                            19  13 004DA        BEQL    67$                                     ;
                              09            58  D1 004DC        CMPL    R8, #9                                  ;
                                            14  13 004DF        BEQL    67$                                     ;
                              01            58  D1 004E1        CMPL    R8, #1                                  ; 1850
                                            0F  13 004E4        BEQL    67$                                     ;
                              0A            58  D1 004E6        CMPL    R8, #10                                 ;
                                            0A  13 004E9        BEQL    67$                                     ;
                              02            58  D1 004EB        CMPL    R8, #2                                  ; 1851
                                            05  13 004EE        BEQL    67$                                     ;
                              03            58  D1 004F0        CMPL    R8, #3                                  ;
                                            3A  12 004F3        BNEQ    71$                                     ;
                              05            58  D1 004F5 67$:   CMPL    R8, #5                                  ; 1854
                                            15  12 004F8        BNEQ    68$                                     ;
                   50 00000000'  EF 00004000  8F  CD 004FA        XORL3   #16384, OPERAND_VALUE, R0               ; 1855
      00000000'  EF               50        FC  8F  9C 00506        ROTL    #-4, R0, OPERAND_VALUE                  ;
                              3F 00000000'  EF  D1 0050F 68$:   CMPL    OPERAND_VALUE, #63                      ; 1856
                                            13  1B 00516        BLEQU   70$                                     ;
                                        0C  BC  DD 00518 69$:   PUSHL   @OPERAND_NUMBER                         ;
                                            01  DD 0051B        PUSHL   #1                                      ;
                                 00028210   8F  DD 0051D        PUSHL   #164368                                 ;
                   00000000G  00            03  FB 00523        CALLS   #3, LIB$SIGNAL                          ;
                                                04 0052A        RET                                             ;
                                            01  DD 0052B 70$:   PUSHL   #1                                      ; 1857
```

```
                                18  11 0052D      BRB      72$                               ;
                                01  DD 0052F 71$: PUSHL    #1                                ; 1861
                    00000000'   EF  9F 00531      PUSHAB   P.AAV                             ;
                          04    AC  DD 00537      PUSHL    ENCODE                            ;
            0000V   CF          03  FB 0053A      CALLS    #3, STORE_OPERAND                 ;
                7E  00000000'EF48   9A 0053F      MOVZBL   DATA_SIZE[R8], -(SP)              ; 1862
                    00000000'   EF  9F 00547 72$: PUSHAB   OPERAND_VALUE                     ;
                          014E  31 0054D          BRW      91$                               ;
                          0C    AE  9F 00550 73$: PUSHAB   ADDRESS                           ; 1871
                                56  DD 00553      PUSHL    LENGTH                            ;
                                57  DD 00555      PUSHL    BUFFER                            ;
                7E              01  CE 00557      MNEGL    #1, -(SP)                         ;
            0000V   CF          04  FB 0055A      CALLS    #4, PARSE_EXPRESSION              ;
                12              50  E8 0055F      BLBS     R0, 74$                           ;
                          0C    BC  DD 00562      PUSHL    @OPERAND_NUMBER                   ;
                                01  DD 00565      PUSHL    #1                                ;
                    00028290    8F  DD 00567      PUSHL    #164496                           ;
        00000000G   00          03  FB 0056D      CALLS    #3, LIB$SIGNAL                    ;
                52        0C    AE  D0 00574 74$: MOVL     ADDRESS, R2                       ; 1872
                                52  DD 00578      PUSHL    R2                                ;
            0000V   CF          01  FB 0057A      CALLS    #1, CHECK_REGISTER                ;
                                50  D5 0057F      TSTL     R0                                ;
                                12  19 00581      BLSS     75$                               ;
                          0C    BC  DD 00583      PUSHL    @OPERAND_NUMBER                   ;
                                01  DD 00586      PUSHL    #1                                ;
                    00028290    8F  DD 00588      PUSHL    #164496                           ;
        00000000G   00          03  FB 0058E      CALLS    #3, LIB$SIGNAL                    ;
        00000100    8F        08    AE  D1 00595 75$: CMPL     R_MODE, #256                  ; 1874
                                17  12 0059D      BNEQ     78$                               ;
        40000000    8F              52  D1 0059F      CMPL     R2, #1073741824               ; 1875
                                06  1F 005A6      BLSSU    76$                               ;
                50        90    8F  9A 005A8      MOVZBL   #144, R0                          ;
                                04  11 005AC      BRB      77$                               ;
                50        E0    8F  9A 005AE 76$: MOVZBL   #224, R0                          ;
            08  AE              50  D0 005B2 77$: MOVL     R0, R_MODE                        ;
                73              5A  E9 005B6 78$: BLBC     DISPLACEMENT_SIZE_NEEDED, 86$     ; 1877
                0F        04    AE  D1 005B9      CMPL     REGNUM, #15                       ; 1878
                                35  12 005BD      BNEQ     80$                               ;
                50        04    AC  D0 005BF      MOVL     ENCODE, R0                        ; 1881
  50      10    A0        08    A0  C1 005C3      ADDL3    8(R0), 16(R0), R0                 ; 1882
                52              50  C2 005C9      SUBL2    R0, R2                            ; 1881
                                52  D7 005CC      DECL     R2                                ; 1880
        FFFFFF81    8F          52  D1 005CE      CMPL     R2, #-127                         ; 1884
                                09  19 005D5      BLSS     79$                               ;
        00000080    8F          52  D1 005D7      CMPL     R2, #128                          ;
                                26  15 005DE      BLEQ     81$                               ;
        FFFF8002    8F          52  D1 005E0 79$: CMPL     R2, #-32766                       ; 1885
                                3B  19 005E7      BLSS     84$                               ;
        00008001    8F          52  D1 005E9      CMPL     R2, #32769                        ;
                                2C  15 005F0      BLEQ     83$                               ;
                                30  11 005F2      BRB      84$                               ; 1886
        FFFFFF80    8F          52  D1 005F4 80$: CMPL     R2, #-128                         ; 1891
                                0F  19 005FB      BLSS     82$                               ;
        0000007F    8F          52  D1 005FD      CMPL     R2, #127                          ;
                                06  14 00604      BGTR     82$                               ;
                50        A0    8F  9A 00606 81$: MOVZBL   #160, R0                          ;
                                1C  11 0060A      BRB      85$                               ;
```

```
                FFFF8000    8F              52  D1 0060C 82$:    CMPL    R2, #-32768                     ; 1892
                                            0F  19 00613         BLSS    84$
                00007FFF    8F              52  D1 00615         CMPL    R2, #32767
                                            06  14 0061C         BGTR    84$
                        50          C0      8F  9A 0061E 83$:    MOVZBL  #192, R0
                                            04  11 00622         BRB     85$
                        50          E0      8F  9A 00624 84$:    MOVZBL  #224, R0                ; 1893
                      08    AE              50  D0 00628 85$:    MOVL    R0, R_MODE              ; 1878
                        51      08    AE    D0  0062C 86$:    MOVL    R_MODE, R1              ; 1896
                00000090    8F              51  D1 00630         CMPL    R1, #144
                                            09  13 00637         BEQL    87$
                00000080    8F              51  D1 00639         CMPL    R1, #128
                                            05  12 00640         BNEQ    88$
                        56                  04  D0 00642 87$:    MOVL    #4, LENGTH
                                            0C  11 00645         BRB     89$
                        50      FF60        C1  9E 00647 88$:    MOVAB   -160(R1), R0            ; 1898
                        50                  20  C6 0064C         DIVL2   #32, R0
                56              01          50  78 0064F         ASHL    R0, #1, LENGTH
                        0F      04    AE    D1  00653 89$:    CMPL    REGNUM, #15             ; 1899
                                            29  12 00657         BNEQ    90$
                00000080    8F              51  D1 00659         CMPL    R1, #128
                                            20  13 00660         BEQL    90$
                00000090    8F              51  D1 00662         CMPL    R1, #144
                                            17  13 00669         BEQL    90$
                        50      04    AC    D0  0066B         MOVL    ENCODE, R0              ; 1900
                52              56      10  A0  C1  0066F         ADDL3   16(R0), LENGTH, R2
                                52      08  A0  C0  00674         ADDL2   8(R0), R2               ; 1901
                50      0C    AE            52  C3  00678         SUBL3   R2, ADDRESS, R0         ; 1900
                        0C    AE      FF    A0  9E  0067D         MOVAB   -1(R0), ADDRESS
                50              51          59  C1  00682 90$:    ADDL3   DEFER, R1, R0           ; 1903
                      08    AE        04 BE40  9E  00686         MOVAB   @REGNUM[R0], R_MODE
                                            01  DD  0068C         PUSHL   #1                      ; 1904
                                        0C  AE  9F  0068E         PUSHAB  R_MODE
                                        04  AC  DD  00691         PUSHL   ENCODE
                0000V   CF                      03  FB  00694         CALLS   #3, STORE_OPERAND
                                                56  DD  00699         PUSHL   LENGTH              ; 1905
                                            10  AE  9F  0069B         PUSHAB  ADDRESS
                                        04  AC  DD  0069E 91$:    PUSHL   ENCODE
                0000V   CF                      03  FB  006A1         CALLS   #3, STORE_OPERAND
                                50      04  AC  D0  006A6         MOVL    ENCODE, R0              ; 1908
                          04    A0              6E  C0  006AA         ADDL2   PARSED, 4(R0)
                                60              6E  A2  006AE         SUBW2   PARSED, (R0)        ; 1909
                                                05  13  006B1         BEQL    92$             ; 1910
                                        04  A0  D6  006B3         INCL    4(R0)               ; 1912
                                            60  B7  006B6         DECW    (R0)                ; 1913
                                                04  006B8 92$:    RET                         ; 1915

; Routine Size:  1721 bytes,    Routine Base:  DBG$CODE + 0B45
```

```
;  1810          1916  1 ROUTINE Parse_Expression(type,string,length,result) =
;  1811          1917  2     BEGIN
;  1812          1918  2
;  1813          1919  2     Routine Get_Result(Input_Desc,Result,Type) =
;  1814          1920  3         BEGIN
;  1815          1921  3
;  1816          1922  3         Routine Handler(sig_args,mch_args) =
;  1817          1923  4             BEGIN
;  1818          1924  4             MAP mch_args : REF BLOCK[,BYTE];
;  1819          1925  4             EXTERNAL ROUTINE Sys$Unwind : Addressing_Mode(General);
;  1820          1926  4             mch_args[chf$l_mch_savr0] = 0;
;  1821          1927  4             Sys$Unwind(0,0);
;  1822          1928  4             RETURN ss$_continue;
;  1823          1929  3             END;                    ! End of routine 'handler'



                                                  0000 00000 HANDLER:.WORD   Save nothing                    ;  1922
                                50        08    AC  D0 00002          MOVL    MCH_ARGS, R0                  ;  1926
                                          0C    A0  D4 00006          CLRL    12(R0)                        ;
                                                7E  7C 00009          CLRQ    -(SP)                         ;  1927
                          00000000G     00        02  FB 0000B          CALLS   #2, SYS$UNWIND                ;
                                50              01  D0 00012          MOVL    #1, R0                        ;  1928
                                                    04 00015          RET                                   ;  1929

; Routine Size:  22 bytes,    Routine Base:  DBG$CODE + 11FE

;  1824          1930  3
;  1825          1931  3         BUILTIN FP;
;  1826          1932  3         LOCAL valdesc : REF dbg$valdesc;
;  1827          1933  3
;  1828          1934  3         .FP = Handler;
;  1829          1935  3         IF (.Type LSS 0) THEN
;  1830          1936  4             BEGIN
;  1831          1937  4             ! Type < 0 means we want an address expression
;  1832          1938  4
;  1833          1939  4             IF NOT DBG$Nparse_Address(.Input_Desc,valdesc,
;  1834          1940  4                 dbg$k_default,token$k_term_none,0) THEN RETURN False;
;  1835          1941  4
;  1836          1942  4             IF NOT DBG$Prim_To_Val(.valdesc,dbg$k_v_value_desc,valdesc)
;  1837          1943  4                 THEN RETURN False;
;  1838          1944  4
;  1839          1945  4             .Result = .valdesc[dbg$l_value_pointer];
;  1840          1946  4             END
;  1841          1947  3         ELSE
;  1842          1948  4             BEGIN
;  1843          1949  4             EXTERNAL dbg$gl_deposit_token;
;  1844          1950  4             LOCAL vms_desc : BLOCK [8,BYTE];
;  1845          1951  4
;  1846          1952  4             ! Type > 0 means we want a value expression
;  1847          1953  4
;  1848          1954  4             IF NOT DBG$Nparse_Expression(.Input_Desc,dbg$k_default,
;  1849          1955  4                         valdesc,token$k_term_none,0) THEN RETURN False;
;  1850          1956  4
```

```
;  1851          1957  4            vms_desc[dsc$b_class]   =  dsc$k_class_s;
;  1852          1958  4            vms_desc[dsc$b_dtype]   = .Data_Type[.Type];
;  1853          1959  4            vms_desc[dsc$w_length]  = .Data_Size[.Type];
;  1854          1960  4            vms_desc[dsc$a_pointer] =  Op_Buffer;
;  1855          1961  4
;  1856          1962  4            Op_Buffer[0,0,32,0] = 0;    ! Clear high operand bytes
;  1857          1963  4            Op_Buffer[4,0,32,0] = 0;    ! to extend to quad value
;  1858          1964  4            DBG$COVER_DX_DX(.valdesc,                                   ! Convert the number
;  1859          1965  4                            dbg$make_val_desc(vms_desc,dbg$k_v_value_desc),
;  1860          1966  4                            false);
;  1861          1967  4 !          DBG$Eval_Lang_Operator(dbg$gl_deposit_token,.valdesc,       ! Delete because if follows l
;  1862          1968  4 !              dbg$make_val_desc(vms_desc,dbg$k_v_value_desc));
;  1863          1969  4
;  1864          1970  3            END;
;  1865          1971  3        RETURN true;
;  1866          1972  2        END;                    ! End of routine 'get_Result'


                                                .EXTRN  DBG$GL_DEPOSIT_TOKEN

                                    000C 00000 GET_RESULT:
                                                .WORD   Save R2,R3                          ; 1919
                    53 00000000'    EF  9E 00002        MOVAB   OP_BUFFER, R3               ;
                    5E              0C  C2 00009        SUBL2   #12, SP                     ;
                    6D        DB    AF  9E 0000C        MOVAB   HANDLER, (FP)               ; 1934
                    52        0C    AC  D0 00010        MOVL    TYPE, R2                    ; 1935
                                    31  18 00014        BGEQ    1$                          ;
                                    7E  7C 00016        CLRQ    -(SP)                       ; 1939
                                    01  DD 00018        PUSHL   #1                          ;
                              0C    AE  9F 0001A        PUSHAB  VALDESC                     ;
                              04    AC  DD 0001D        PUSHL   INPUT_DESC                  ;
                    00000000G 00        05  FB 00020    CALLS   #5, DBG$NPARSE_ADDRESS      ;
                              6D        50  E9 00027    BLBC    R0, 3$                      ;
                                        5E  DD 0002A    PUSHL   SP                          ; 1942
                              7E  83    8F  9A 0002C    MOVZBL  #131, -(SP)                 ;
                              08        AE  DD 00030    PUSHL   VALDESC                     ;
                    00000000G 00        03  FB 00033    CALLS   #3, DBG$PRIM_TO_VAL         ;
                              5A        50  E9 0003A    BLBC    R0, 3$                      ;
                              50        6E  D0 0003D    MOVL    VALDESC, R0                 ; 1945
                        08    BC  18    A0  D0 00040    MOVL    24(R0), @RESULT             ;
                                        4C  11 00045    BRB     2$                          ; 1935
                                        7E  7C 00047 1$:    CLRQ    -(SP)                   ; 1954
                              08        AE  9F 00049    PUSHAB  VALDESC                     ;
                                        01  DD 0004C    PUSHL   #1                          ;
                              04        AC  DD 0004E    PUSHL   INPUT_DESC                  ;
                    00000000G 00        05  FB 00051    CALLS   #5, DBG$NPARSE_EXPRESSION   ;
                              3C        50  E9 00058    BLBC    R0, 3$                      ;
                        07    AE            01  90 0005B    MOVB    #1, VMS_DESC+3          ; 1957
                        06    AE  00000000'EF42 90 0005F    MOVB    DATA_TYPE[R2], VMS_DESC+2   ; 1958
                        04    AE  00000000'EF42 9B 00068    MOVZBW  DATA_SIZE[R2], VMS_DESC     ; 1959
                        08    AE            63  9E 00071    MOVAB   OP_BUFFER, VMS_DESC+4       ; 1960
                                            63  7C 00075    CLRQ    OP_BUFFER               ; 1962
                                            7E  D4 00077    CLRL    -(SP)                   ; 1964
                                  7E    83    8F  9A 00079    MOVZBL  #131, -(SP)             ; 1965
                                        0C  AE  9F 0007D    PUSHAB  VMS_DESC                ;
                    00000000G 00            02  FB 00080    CALLS   #2, DBG$MAKE_VAL_DESC   ;
```

```
                                    50  DD 00087          PUSHL   R0                                      ;
                                08  AE  DD 00089          PUSHL   VALDESC                                 ; 1964
                00000000G   00      03  FB 0008C          CALLS   #3, DBG$COVER_DX_DX                     ;
                            50      01  D0 00093 2$:      MOVL    #1, R0                                  ; 1971
                                    04  00096            RET                                             ;
                                    50  D4 00097 3$:      CLRL    R0                                      ; 1972
                                    04  00099            RET                                             ;

; Routine Size:  154 bytes,    Routine Base:  DBG$CODE + 1214

; 1867          1973  2
; 1868          1974  2     LOCAL
; 1869          1975  2         Mark,Status,
; 1870          1976  2         Term_Char       : BYTE UNSIGNED,
; 1871          1977  2         Term_Addr       : REF VECTOR [1,BYTE],
; 1872          1978  2         Local_Desc      : BLOCK [8,BYTE];
; 1873          1979  2
; 1874          1980  2     Local_Desc[dsc$b_class]   = dsc$k_class_s;
; 1875          1981  2     Local_Desc[dsc$b_dtype]   = dsc$k_dtype_t;
; 1876          1982  2     Local_Desc[dsc$w_length]  = .length;
; 1877          1983  2     Local_Desc[dsc$a_pointer] = .string;
; 1878          1984  2
; 1879          1985  2     Mark = dbg$push_tempmem();
; 1880          1986  2     Term_Addr = .string + .length;
; 1881          1987  2     Term_Char = .Term_Addr[0];
; 1882          1988  2     Term_Addr[0] = 13;
; 1883          1989  2     Status = Get_Result(Local_Desc,.Result,.Type);
; 1884          1990  2     Term_Addr[0] = .Term_Char;
; 1885          1991  2     dbg$pop_tempmem(.Mark);
; 1886          1992  2     RETURN .Status;
; 1887          1993  1     END;



                                        003C 00000 PARSE_EXPRESSION:
                                                    .WORD   Save R2,R3,R4,R5                            ; 1916
                            5E                  08  C2 00002            SUBL2   #8, SP                  ;
                        02  AE          010E    8F  B0 00005            MOVW    #270, LOCAL_DESC+2      ; 1981
                            6E              0C  AC  B0 0000B            MOVW    LENGTH, LOCAL_DESC      ; 1982
                        04  AE              08  AC  D0 0000F            MOVL    STRING, LOCAL_DESC+4    ; 1983
                00000000G   00                  00  FB 00014            CALLS   #0, DBG$PUSH_TEMPMEM    ; 1985
                            55                  50  D0 0001B            MOVL    R0, MARK                ;
                52      08  AC              0C  AC  C1 0001E            ADDL3   LENGTH, STRING, TERM_ADDR ; 1986
                            53                  62  90 00024            MOVB    (TERM_ADDR), TERM_CHAR  ; 1987
                            62                  0D  90 00027            MOVB    #13, (TERM_ADDR)        ; 1988
                                            04  AC  DD 0002A            PUSHL   TYPE                    ; 1989
                                            10  AC  DD 0002D            PUSHL   RESULT                  ;
                                            08  AE  9F 00030            PUSHAB  LOCAL_DESC              ;
                            FF2E            CF  03  FB 00033            CALLS   #3, GET_RESULT          ;
                                            54  50  D0 00038            MOVL    R0, STATUS              ;
                                            62  53  90 0003B            MOVB    TERM_CHAR, (TERM_ADDR)  ; 1990
                                                55  DD 0003E            PUSHL   MARK                    ; 1991
                00000000G   00                  01  FB 00040            CALLS   #1, DBG$POP_TEMPMEM     ;
                            50                  54  D0 00047            MOVL    STATUS, R0              ; 1992
```

```
                                04 0004A        RET                                     ; 1993

; Routine Size:  75 bytes,    Routine Base:  DBG$CODE + 12AE
```

```
;1889     1994  1 ROUTINE Parse_Register(string,length) =
;1890     1995  2     BEGIN
;1891     1996  2     BIND
;1892     1997  2         Two_char_register = .string + .length - 2 : BLOCKVECTOR [1,WORD],      The last 2 characters
;1893     1998  2         Three_char_register = .string + .length - 3 : BLOCKVECTOR [1,3];       The last 3 characters
;1894     1999  2
;1895     2000  2     IF .length LSS 2                                                           There must be at least 2 ch
;1896     2001  2     THEN
;1897     2002  2         RETURN -1;
;1898     2003  2
;1899     2004  2     SELECTONE .Two_char_register[ 0, 0, 0, 16, 0 ] OF                          First character
;1900     2005  2         SET
;1901     2006  2         [ .UPLIT('R0') ]:       RETURN 0;
;1902     2007  2         [ .UPLIT('R1') ]:       RETURN 1;
;1903     2008  2         [ .UPLIT('R2') ]:       RETURN 2;
;1904     2009  2         [ .UPLIT('R3') ]:       RETURN 3;
;1905     2010  2         [ .UPLIT('R4') ]:       RETURN 4;
;1906     2011  2         [ .UPLIT('R5') ]:       RETURN 5;
;1907     2012  2         [ .UPLIT('R6') ]:       RETURN 6;
;1908     2013  2         [ .UPLIT('R7') ]:       RETURN 7;
;1909     2014  2         [ .UPLIT('R8') ]:       RETURN 8;
;1910     2015  2         [ .UPLIT('R9') ]:       RETURN 9;
;1911     2016  2         [ .UPLIT('AP') ]:       RETURN 12;
;1912     2017  2         [ .UPLIT('FP') ]:       RETURN 13;
;1913     2018  2         [ .UPLIT('SP') ]:       RETURN 14;
;1914     2019  2         [ .UPLIT('PC') ]:       RETURN 15;
;1915     2020  2         [ OTHERWISE ]:                                                         Might be a 3 character regi
;1916     2021  3             BEGIN
;1917     2022  3             IF .length LSS 3                                                   There must be at least 3 ch
;1918     2023  3             THEN
;1919     2024  3                 RETURN -1;
;1920     2025  3
;1921     2026  3             SELECTONE .Three_char_register[ 0, 0, 0, 24, 0 ] OF
;1922     2027  3                 SET
;1923     2028  3                 [ .UPLIT('R10') ]:          RETURN 10;
;1924     2029  3                 [ .UPLIT('R11') ]:          RETURN 11;
;1925     2030  3                 [ .UPLIT('R12') ]:          RETURN 12;
;1926     2031  3                 [ .UPLIT('R13') ]:          RETURN 13;
;1927     2032  3                 [ .UPLIT('R14') ]:          RETURN 14;
;1928     2033  3                 [ .UPLIT('R15') ]:          RETURN 15;
;1929     2034  3                 [ OTHERWISE ]:              RETURN -1;
;1930     2035  3                 TES;
;1931     2036  2             END;
;1932     2037  2         TES;
;1933     2038  1     END;


                                                    .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

                                00  00  30  52  011D0 P.AAW:  .ASCII  \R0\<0><0>                  ;
                                00  00  31  52  011D4 P.AAX:  .ASCII  \R1\<0><0>                  ;
                                00  00  32  52  011D8 P.AAY:  .ASCII  \R2\<0><0>                  ;
                                00  00  33  52  011DC P.AAZ:  .ASCII  \R3\<0><0>                  ;
                                00  00  34  52  011E0 P.ABA:  .ASCII  \R4\<0><0>                  ;
                                00  00  35  52  011E4 P.ABB:  .ASCII  \R5\<0><0>                  ;
                                00  00  36  52  011E8 P.ABC:  .ASCII  \R6\<0><0>                  ;
```

```
                        00  00  37  52  011EC P.ABD:  .ASCII  \R7\<0><0>                    ;
                        00  00  38  52  011F0 P.ABE:  .ASCII  \R8\<0><0>                    ;
                        00  00  39  52  011F4 P.ABF:  .ASCII  \R9\<0><0>                    ;
                        00  00  50  41  011F8 P.ABG:  .ASCII  \AP\<0><0>                    ;
                        00  00  50  46  011FC P.ABH:  .ASCII  \FP\<0><0>                    ;
                        00  00  50  53  01200 P.ABI:  .ASCII  \SP\<0><0>                    ;
                        00  00  43  50  01204 P.ABJ:  .ASCII  \PC\<0><0>                    ;
                        00  30  31  52  01208 P.ABK:  .ASCII  \R10\<0>                      ;
                        00  31  31  52  0120C P.ABL:  .ASCII  \R11\<0>                      ;
                        00  32  31  52  01210 P.ABM:  .ASCII  \R12\<0>                      ;
                        00  33  31  52  01214 P.ABN:  .ASCII  \R13\<0>                      ;
                        00  34  31  52  01218 P.ABO:  .ASCII  \R14\<0>                      ;
                        00  35  31  52  0121C P.ABP:  .ASCII  \R15\<0>                      ;


                                                      .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                                    0004 00000 PARSE_REGISTER:
                                                      .WORD   Save R2                     ; 1994
                    52 00000000'  EF  9E 00002        MOVAB   P.AAW, R2                   ;
51      04  AC      08  AC  C1 00009        ADDL3   LENGTH, STRING, R1          ; 1997
            02      08  AC  D1 0000F        CMPL    LENGTH, #2                  ; 2000
                        03  18 00013        BGEQ    1$                          ;
                    00C6    31 00015        BRW     21$                         ;
                    50      FE  A1  3C 00018 1$:     MOVZWL  -2(R1), R0          ; 2004
                    62          50  D1 0001C        CMPL    R0, P.AAW           ; 2006
                                03  12 0001F        BNEQ    2$                  ;
                                50  D4 00021        CLRL    R0                  ;
                                    04 00023        RET                         ;
                04  A2          50  D1 00024 2$:     CMPL    R0, P.AAX           ; 2007
                                04  12 00028        BNEQ    3$                  ;
                    50          01  D0 0002A        MOVL    #1, R0              ;
                                    04 0002D        RET                         ;
                08  A2          50  D1 0002E 3$:     CMPL    R0, P.AAY           ; 2008
                                04  12 00032        BNEQ    4$                  ;
                    50          02  D0 00034        MOVL    #2, R0              ;
                                    04 00037        RET                         ;
                0C  A2          50  D1 00038 4$:     CMPL    R0, P.AAZ           ; 2009
                                04  12 0003C        BNEQ    5$                  ;
                    50          03  D0 0003E        MOVL    #3, R0              ;
                                    04 00041        RET                         ;
                10  A2          50  D1 00042 5$:     CMPL    R0, P.ABA           ; 2010
                                04  12 00046        BNEQ    6$                  ;
                    50          04  D0 00048        MOVL    #4, R0              ;
                                    04 0004B        RET                         ;
                14  A2          50  D1 0004C 6$:     CMPL    R0, P.ABB           ; 2011
                                04  12 00050        BNEQ    7$                  ;
                    50          05  D0 00052        MOVL    #5, R0              ;
                                    04 00055        RET                         ;
                18  A2          50  D1 00056 7$:     CMPL    R0, P.ABC           ; 2012
                                04  12 0005A        BNEQ    8$                  ;
                    50          06  D0 0005C        MOVL    #6, R0              ;
                                    04 0005F        RET                         ;
                1C  A2          50  D1 00060 8$:     CMPL    R0, P.ABD           ; 2013
                                04  12 00064        BNEQ    9$                  ;
                    50          07  D0 00066        MOVL    #7, R0              ;
```

```
                                                       04 00069        RET                                          ;
                                20  A2            50   D1 0006A  9$:   CMPL     R0, P.ABE                           ; 2014
                                                  04   12 0006E        BNEQ     10$                                 ;
                                    50            08   D0 00070        MOVL     #8, R0                              ;
                                                       04 00073        RET                                          ;
                                24  A2            50   D1 00074  10$:  CMPL     R0, P.ABF                           ; 2015
                                                  04   12 00078        BNEQ     11$                                 ;
                                    50            09   D0 0007A        MOVL     #9, R0                              ;
                                                       04 0007D        RET                                          ;
                                28  A2            50   D1 0007E  11$:  CMPL     R0, P.ABG                           ; 2016
                                                  38   13 00082        BEQL     14$                                 ;
                                2C  A2            50   D1 00084        CMPL     R0, P.ABH                           ; 2017
                                                  3C   13 00088        BEQL     16$                                 ;
                                30  A2            50   D1 0008A        CMPL     R0, P.ABI                           ; 2018
                                                  40   13 0008E        BEQL     18$                                 ;
                                34  A2            50   D1 00090        CMPL     R0, P.ABJ                           ; 2019
                                                  44   13 00094        BEQL     20$                                 ;
                                    03     08     AC   D1 00096        CMPL     LENGTH, #3                          ; 2022
                                                  42   19 0009A        BLSS     21$                                 ;
            50      FD   A1             18        00   EF 0009C        EXTZV    #0, #24, -3(R1), R0                 ; 2026
                                38  A2            50   D1 000A2        CMPL     R0, P.ABK                           ; 2028
                                                  04   12 000A6        BNEQ     12$                                 ;
                                    50            0A   D0 000A8        MOVL     #10, R0                             ;
                                                       04 000AB        RET                                          ;
                                3C  A2            50   D1 000AC  12$:  CMPL     R0, P.ABL                           ; 2029
                                                  04   12 000B0        BNEQ     13$                                 ;
                                    50            0B   D0 000B2        MOVL     #11, R0                             ;
                                                       04 000B5        RET                                          ;
                                40  A2            50   D1 000B6  13$:  CMPL     R0, P.ABM                           ; 2030
                                                  04   12 000BA        BNEQ     15$                                 ;
                                    50            0C   D0 000BC  14$:  MOVL     #12, R0                             ;
                                                       04 000BF        RET                                          ;
                                44  A2            50   D1 000C0  15$:  CMPL     R0, P.ABN                           ; 2031
                                                  04   12 000C4        BNEQ     17$                                 ;
                                    50            0D   D0 000C6  16$:  MOVL     #13, R0                             ;
                                                       04 000C9        RET                                          ;
                                48  A2            50   D1 000CA  17$:  CMPL     R0, P.ABO                           ; 2032
                                                  04   12 000CE        BNEQ     19$                                 ;
                                    50            0E   D0 000D0  18$:  MOVL     #14, R0                             ;
                                                       04 000D3        RET                                          ;
                                4C  A2            50   D1 000D4  19$:  CMPL     R0, P.ABP                           ; 2033
                                                  04   12 000D8        BNEQ     21$                                 ;
                                    50            0F   D0 000DA  20$:  MOVL     #15, R0                             ;
                                                       04 000DD        RET                                          ;
                                    50            01   CE 000DE  21$:  MNEGL    #1, R0                              ; 2034
                                                       04 000E1        RET                                          ; 2038

; Routine Size:  226 bytes,    Routine Base:  DBG$CODE + 12F9


; 1934          2039  1
; 1935          2040  1
; 1936          2041  1
; 1937          2042  1
; 1938          2043  1
; 1939          2044  1 ROUTINE Check_Register(Address : UNSIGNED) =
; 1940          2045  2     BEGIN
```

```
;  1941          2046  2     EXTERNAL dbg$reg_values;
;  1942          2047  2     Address = .Address - dbg$reg_values;        ! ** NOTE NO "."
;  1943          2048  2     IF (.Address GTRU 15*%UPVAL) THEN RETURN -1;
;  1944          2049  2     IF ((.Address AND (%UPVAL-1)) NEQ 0) THEN RETURN -1;
;  1945          2050  2     RETURN .Address/%UPVAL;
;  1946          2051  1     END;


                                                        .EXTRN  DBG$REG_VALUES

                                        0000 00000 CHECK_REGISTER:
                                                        .WORD   Save nothing                    ;  2044
                     50 00000000G   00  9E 00002        MOVAB   DBG$REG_VALUES, R0              ;  2047
                04   AC             50  C2 00009        SUBL2   R0, ADDRESS                     ;
                     3C        04   AC  D1 0000D        CMPL    ADDRESS, #60                    ;  2048
                               06   1A 00011            BGTRU   1$                              ;
                     03        04   AC  93 00013        BITB    ADDRESS, #3                     ;  2049
                               04   13 00017            BEQL    2$                              ;
                     50             01  CE 00019 1$:    MNEGL   #1, R0                          ;
                                    04 0001C            RET                                     ;
                50   04   AC        04  C7 0001D 2$:    DIVL3   #4, ADDRESS, R0                 ;  2050
                                    04 00022            RET                                     ;  2051

; Routine Size:  35 bytes,    Routine Base:  DBG$CODE + 13DB
```

```
1 ROUTINE Store_Operand(EnCode,Operand,Length) : NOVALUE =
2     BEGIN
2     MAP
2         EnCode  : REF BLOCK [20,BYTE] FIELD(EnCode_Fields),
2         Operand : REF VECTOR [ ,BYTE];
2
2     ch$move(.Length, Operand[0],
2             .EnCode[Enc_Output_Length] + .EnCode[Enc_Output_Buffer]);
2
2     EnCode[Enc_Output_Length] = .EnCode[Enc_Output_Length] + .Length;
1     END;
```

```
                                      007C 00000 STORE_OPERAND:
                                                        .WORD   Save R2,R3,R4,R5,R6                     ; 2052
                    56         04    AC   D0 00002      MOVL    ENCODE, R6                              ; 2059
        50    08    A6         0C    A6   C1 00006      ADDL3   12(R6), 8(R6), R0                       ;
        60    08    BC         0C    AC   28 0000C      MOVC3   LENGTH, @OPERAND, (R0)                  ;
              08    A6         0C    AC   C0 00012      ADDL2   LENGTH, 8(R6)                           ; 2061
                                          04 00017      RET                                             ; 2062
```

; Routine Size: 24 bytes, Routine Base: DBG$CODE + 13FE

```
;  1960          2063  1 ROUTINE Print_Address(Address) : NOVALUE =
;  1961          2064  2     BEGIN
;  1962          2065  2     LOCAL mark;
;  1963          2066  2     mark = DBG$Push_Tempmem();
;  1964          2067  2     DBG$Print_Identifier_PC(.Address);
;  1965          2068  2     DBG$Pop_Tempmem(.mark);
;  1966          2069  1     END;


                                        0004 00000 PRINT_ADDRESS:
                                                   .WORD   Save R2                                  ; 2063
                    00000000G  00           00  FB 00002    CALLS   #0, DBG$PUSH_TEMPMEM             ; 2066
                               52           50  D0 00009    MOVL    R0, MARK                         ;
                                         04    AC  DD 0000C    PUSHL   ADDRESS                      ; 2067
                    00000000G  00           01  FB 0000F    CALLS   #1, DBG$PRINT_IDENTIFIER_PC      ;
                                            52  DD 00016    PUSHL   MARK                         ; 2068
                    00000000G  00           01  FB 00018    CALLS   #1, DBG$POP_TEMPMEM              ;
                                            04  0001F      RET                                      ; 2069

; Routine Size:  32 bytes,    Routine Base:  DBG$CODE + 1416


;  1967          2070  1
;  1968          2071  1 ROUTINE Print_Operand(Context) : NOVALUE =
;  1969          2072  2     BEGIN
;  1970          2073  2     LOCAL
;  1971          2074  2         mark,
;  1972          2075  2         vms_desc        : dbg$stg_desc;
;  1973          2076  2
;  1974          2077  2     mark = DBG$Push_Tempmem();
;  1975          2078  2     vms_desc[dsc$b_class]   = dsc$k_class_s;
;  1976          2079  2     vms_desc[dsc$b_dtype]   = .data_type[.context];
;  1977          2080  2     vms_desc[dsc$w_length]  = .data_size[.context];
;  1978          2081  2     vms_desc[dsc$a_pointer] = Op_Buffer;
;  1979          2082  2     DBG$Print_Value(DBG$Make_Val_Desc(vms_desc,dbg$k_value_desc),dbg$k_default,False,False);
;  1980          2083  2     DBG$Pop_Tempmem(.mark);
;  1981          2084  1     END;


                                        0004 00000 PRINT_OPERAND:
                                                   .WORD   Save R2                                  ; 2071
                               5E           0C  C2 00002    SUBL2   #12, SP                          ;
                    00000000G  00           00  FB 00005    CALLS   #0, DBG$PUSH_TEMPMEM             ; 2077
                               52           50  D0 0000C    MOVL    R0, MARK                         ;
                         03    AE           01  90 0000F    MOVB    #1, VMS_DESC+3                   ; 2078
                               50 00000000' EF  9E 00013    MOVAB   DATA_TYPE, R0                    ; 2079
                         02    AE       04 BC40  90 0001A    MOVB    @CONTEXT[R0], VMS_DESC+2         ;
                               50 00000000' EF  9E 00020    MOVAB   DATA_SIZE, R0                    ; 2080
                               6E       04 BC40  9B 00027    MOVZBW  @CONTEXT[R0], VMS_DESC           ;
                         04    AE 00000000' EF  9E 0002C    MOVAB   OP_BUFFER, VMS_DESC+4            ; 2081
                                            7E  7C 00034    CLRQ    -(SP)                            ; 2082
```

```
                                 01  DD 00036          PUSHL   #1
                          7E     7A  8F  9A 00038      MOVZBL  #122, -(SP)
                                 10  AE  9F 0003C      PUSHAB  VMS_DESC
                00000000G   00     02  FB 0003F      CALLS   #2, DBG$MAKE_VAL_DESC
                                   50  DD 00046      PUSHL   R0
                00000000G   00     04  FB 00048      CALLS   #4, DBG$PRINT_VALUE
                                   52  DD 0004F      PUSHL   MARK                        ;2083
                00000000G   00     01  FB 00051      CALLS   #1, DBG$POP_TEMPMEM
                                       04 00058      RET                                 ;2084

; Routine Size:  89 bytes,    Routine Base:  DBG$CODE + 1436
```

```
;  1983          2085  1 ROUTINE Scan_Operand(Input_Desc : REF dbg$stg_desc,Delimiter) =
;  1984          2086  2     BEGIN
;  1985          2087  2     BUILTIN ACTUALCOUNT,ACTUALPARAMETER;
;  1986          2088  2     LOCAL
;  1987          2089  2         Depth,
;  1988          2090  2         Local_Desc  : dbg$stg_desc,
;  1989          2091  2         char        : BYTE UNSIGNED;
;  1990          2092  2
;  1991          2093  2     Depth = (IF Actualcount() LSS 3 THEN 0 ELSE Actualparameter(3)+1);
;  1992          2094  2
;  1993          2095  2     Skip_Leading_Blanks(.Input_Desc);
;  1994          2096  2
;  1995          2097  2     ch$move(8,Input_Desc[0,0,0,0],Local_Desc[0,0,0,0]);
;  1996          2098  2
;  1997          2099  2     WHILE .Local_Desc[dsc$w_length] GEQ 0 DO
;  1998          2100  3         BEGIN
;  1999          2101  3         LOCAL Target,Length;
;  2000          2102  4         char = (IF .Local_Desc[dsc$w_length] EQL 0 THEN 0
;  2001          2103  3                 ELSE .(.Local_Desc[dsc$a_pointer])<0,8,0>);
;  2002          2104  3         Target = 0;
;  2003          2105  3         IF .char EQL %X'09' THEN (.Local_Desc[dsc$a_pointer])<0,8,0> = char = %C' ';
;  2004          2106  3         IF .char EQL .Delimiter THEN
;  2005          2107  4           RETURN (.Local_Desc[dsc$a_pointer] - .Input_Desc[dsc$a_pointer])
;  2006          2108  3         ELSE IF .char EQL  0  THEN EXITLOOP;
;  2007          2109  3         IF .Delimiter NEQ %C'''' THEN
;  2008          2110  4             BEGIN
;  2009          2111  5             IF (.char GEQ %C'a') AND (.char LEQ %C'z')
;  2010          2112  5             THEN (.Local_Desc[dsc$a_pointer])<0,8,0> = .char - (%C'a'-%C'A')
;  2011          2113  4             ELSE IF .char EQL %C'''' THEN Target = %C''''
;  2012          2114  4             ELSE IF .char EQL %C'(' THEN Target = %C')'
;  2013          2115  4             ELSE IF .char EQL %C'[' THEN Target = %C']';
;  2014          2116  3             END;
;  2015          2117  3         Local_Desc[dsc$w_length]  = .Local_Desc[dsc$w_length]  - 1;
;  2016          2118  3         Local_Desc[dsc$a_pointer] = .Local_Desc[dsc$a_pointer] + 1;
;  2017          2119  3         IF .Target NEQ 0 THEN
;  2018          2120  4             BEGIN
;  2019          2121  4             Length = Scan_Operand(Local_Desc,.Target,.Depth) + 1;
;  2020          2122  4             Local_Desc[dsc$w_length]  = .Local_Desc[dsc$w_length] -.Length;
;  2021          2123  4             Local_Desc[dsc$a_pointer] = .Local_Desc[dsc$a_pointer]+.Length;
;  2022          2124  3             END;
;  2023          2125  2         END;
;  INFO#252                 L1:2099
;  Test expression is always true
;  2024          2126  2     IF .Depth NEQ 0 THEN SIGNAL(dbg$_nodelimtr);
;  2025          2127  2     RETURN .Input_Desc[dsc$w_length];
;  2026          2128  1     END;


                                    00FC 00000 SCAN_OPERAND:
                                                       .WORD   Save R2,R3,R4,R5,R6,R7                  ; 2085
                                5E                  0C  C2 00002          SUBL2   #12, SP                 ;
                                03                  6C  91 00005          CMPB    (AP), #3                ; 2093
                                                    04  1E 00008          BGEQU   1$                      ;
                                                    57  D4 0000A          CLRL    DEPTH                   ;
```

```
                                       05 11 0000C        BRB     2$                              ;
              57      0C    AC         01 C1 0000E 1$:    ADDL3   #1, 12(AP), DEPTH               ;
                            56     04  AC D0 00013 2$:    MOVL    INPUT_DESC, R6                  ; 2095
                                       56 DD 00017        PUSHL   R6                              ;
                   0000V    CF         01 FB 00019        CALLS   #1, SKIP_LEADING_BLANKS         ;
              6E            66         08 28 0001E        MOVC3   #8, (R6), LOCAL_DESC            ; 2097
                                       6E B5 00022 3$:    TSTW    LOCAL_DESC                      ; 2102
                                       04 12 00024        BNEQ    4$                              ;
                                       50 D4 00026        CLRL    R0                              ;
                                       04 11 00028        BRB     5$                              ;
                            50     04  BE 9A 0002A 4$:    MOVZBL  @LOCAL_DESC+4, R0               ; 2103
                            52         50 90 0002E 5$:    MOVB    R0, CHAR                        ; 2102
                                       51 D4 00031        CLRL    TARGET                          ; 2104
                            09         52 91 00033        CMPB    CHAR, #9                        ; 2105
                                       07 12 00036        BNEQ    6$                              ;
                            52         20 90 00038        MOVB    #32, CHAR                       ;
                      04    BE         20 90 0003B        MOVB    #32, @LOCAL_DESC+4              ;
    08   AC   52            08         00 ED 0003F 6$:    CMPZV   #0, #8, CHAR, DELIMITER         ; 2106
                                       07 12 00045        BNEQ    7$                              ;
              50      04    AE     04  A6 C3 00047        SUBL3   4(R6), LOCAL_DESC+4, R0         ; 2107
                                       04 0004D           RET                                     ;
                                       52 95 0004E 7$:    TSTB    CHAR                            ; 2108
                                       58 13 00050        BEQL    12$                             ;
                            22     08  AC D1 00052        CMPL    DELIMITER, #34                  ; 2109
                                       31 13 00056        BEQL    11$                             ;
                      61    8F         52 91 00058        CMPB    CHAR, #97                       ; 2111
                                       0D 1F 0005C        BLSSU   8$                              ;
                      7A    8F         52 91 0005E        CMPB    CHAR, #122                      ;
                                       07 1A 00062        BGTRU   8$                              ;
         04   BE            52         20 83 00064        SUBB3   #32, CHAR, @LOCAL_DESC+4        ; 2112
                                       1E 11 00069        BRB     11$                             ;
                            22         52 91 0006B 8$:    CMPB    CHAR, #34                       ; 2113
                                       05 12 0006E        BNEQ    9$                              ;
                            51         22 D0 00070        MOVL    #34, TARGET                     ;
                                       14 11 00073        BRB     11$                             ;
                            28         52 91 00075 9$:    CMPB    CHAR, #40                       ; 2114
                                       05 12 00078        BNEQ    10$                             ;
                            51         29 D0 0007A        MOVL    #41, TARGET                     ;
                                       0A 11 0007D        BRB     11$                             ;
                      5B    8F         52 91 0007F 10$:   CMPB    CHAR, #91                       ; 2115
                                       04 12 00083        BNEQ    11$                             ;
                            51     5D  8F 9A 00085        MOVZBL  #93, TARGET                     ;
                                       6E B7 00089 11$:   DECW    LOCAL_DESC                      ; 2117
                                   04  AE D6 0008B        INCL    LOCAL_DESC+4                    ; 2118
                                       51 D5 0008E        TSTL    TARGET                          ; 2119
                                       90 13 00090        BEQL    3$                              ;
                                 0082  8F BB 00092        PUSHR   #^M<R1,R7>                      ; 2121
                                   08  AE 9F 00096        PUSHAB  LOCAL_DESC                      ;
                    FF62    CF         03 FB 00099        CALLS   #3, SCAN_OPERAND                ;
                                       50 D6 0009E        INCL    LENGTH                          ;
                            6E         50 A2 000A0        SUBW2   LENGTH, LOCAL_DESC              ; 2122
                      04    AE         50 C0 000A3        ADDL2   LENGTH, LOCAL_DESC+4            ; 2123
                                     FF78 31 000A7        BRW     3$                              ; 2099
                                       57 D5 000AA 12$:   TSTL    DEPTH                           ; 2126
                                       0D 13 000AC        BEQL    13$                             ;
                             00028218  8F DD 000AE        PUSHL   #164376                         ;
               00000000G    00         01 FB 000B4        CALLS   #1, LIB$SIGNAL                  ;
```

```
                                50          66  3C 000BB 13$:   MOVZWL  (R6), R0                        ; 2127
                                                04 000BE        RET                                     ; 2128

; Routine Size:  191 bytes,    Routine Base:  DBG$CODE + 148F
```

```
;  2028          2129  1 ROUTINE Skip_Leading_Blanks(Input_Desc : REF dbg$stg_desc) : NOVALUE =
;  2029          2130  2     BEGIN
;  2030          2131  2     WHILE .Input_Desc[dsc$w_length] GTR 0 DO
;  2031          2132  3         BEGIN
;  2032          2133  3         LOCAL char : BYTE UNSIGNED;
;  2033          2134  3         char = .(.Input_Desc[dsc$a_pointer])<0,8,0>;
;  2034          2135  3         IF (.char NEQ %X'20') AND (.char NEQ %X'09') THEN EXITLOOP;
;  2035          2136  3         Input_Desc[dsc$w_length]  = .Input_Desc[dsc$w_length]  - 1;
;  2036          2137  3         Input_Desc[dsc$a_pointer] = .Input_Desc[dsc$a_pointer] + 1;
;  2037          2138  2         END;
;  2038          2139  1     END;


                                    0000 00000 SKIP_LEADING_BLANKS:
                                                       .WORD   Save nothing                    ;  2129
                          50    04  AC  D0 00002         MOVL    INPUT_DESC, R0              ;  2134
                                04  BC  B5 00006 1$:      TSTW    @INPUT_DESC                 ;  2131
                                    16  13 00009         BEQL    3$                          ;
                          51    04  B0  90 0000B         MOVB    @4(R0), CHAR                ;  2134
                          20        51  91 0000F         CMPB    CHAR, #32                   ;  2135
                                    05  13 00012         BEQL    2$                          ;
                          09        51  91 00014         CMPB    CHAR, #9                    ;
                                    08  12 00017         BNEQ    3$                          ;
                                04  BC  B7 00019 2$:      DECW    @INPUT_DESC                 ;  2136
                                04  A0  D6 0001C         INCL    4(R0)                       ;  2137
                                    E5  11 0001F         BRB     1$                          ;  2131
                                        04 00021 3$:     RET                                 ;  2139

; Routine Size:  34 bytes,    Routine Base:  DBG$CODE + 154E

;  2039          2140  1
;  2040          2141  1 END
;  2041          2142  0 ELUDOM


                                                       .EXTRN  LIB$SIGNAL

;                                  PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  DBG$PLIT                      4640  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(0)
;  DBG$OWN                         16  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  PEL,  CON,  PIC,ALIGN(2)
;  DBG$CODE                      5488  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(0)


;                                Library Statistics
;
;                                        -------- Symbols --------      Pages       Processing
```

```
;       File                                    Total     Loaded   Percent      Mapped       Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1                18619         22         0        1000      00:01.9
;  _$255$DUA28:[DEBUG.OBJ]STRUCDEF.L32;1           32          0         0           7      00:00.1
;  _$255$DUA28:[DEBUG.OBJ]DBGLIB.L32;1           1545         56         3          97      00:01.9
;  _$255$DUA28:[DEBUG.OBJ]DSTRECRDS.L32;1
;                                                 418          0         0          31      00:00.3
;  _$255$DUA28:[DEBUG.OBJ]DBGMSG.L32;1            386         15         3          22      00:00.3



; Information:  1
; Warnings:     0
; Errors:       0


;                                       COMMAND QUALIFIERS

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

; Size:          5488 code + 4656 data bytes
; Run Time:         02:23.4
; Elapsed Time:     07:22.8
; Lines/CPU Min:      896
; Lexemes/CPU-Min: 43556
; Memory Used:  668 pages
; Compilation Complete
```

0080 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

DBGEVALOP
LIS

DBGENCDEC
LIS