HIGH SPEED TEXT SEARCH SYSTEM

HSTS SOFTWARE
LISTINGS

VOL. 4 OF 5

Diagnostics
Part I

STAT

HSTS MASTER COMPUTER SOFTWARE LISTINGS

SL120100

VOLUME 4 of 5

Prepared for:

Central Intelligence Agency
Washington, DC 20505

[redacted] R80-016

STAT

March 1980

STAT

DIAGNOSTICS
PART 1

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200020001-3

```
                        .TITLE··TMT·
                        .LIST·  MEB·
                ;
                ;
                ;       TERM·DETECTOR·MEMORY TESTS
                ;       MAIN·MODULE·
                ;
                ;
                ;       THIS·MODULE·EXECUTES·MEMORY·TEST CYCLES. ALL·CONTROL
                ;       INFORMATION·FOR·A·TEST CYCLE·IS·OBTAINED·FROM·THE·
                ;       COMMAND·LINE. TMT·BUILDS·TABLES·AND·SETS·FLAGS·IN·
                ;       ORDER·THAT·THERE·NEED·BE·NO·USER INTERVENTION·DURING
                ;       THE·EXECUTION·OF·THE·CYCLE. THE·PROMPTS·FOR·COMMAND·
                ;       LINE·INPUT·AND·GENERAL TMT·ACTION·ARE:
                ;
                ;       1. PROMPT·FOR·ALL·TESTS, ALL·MEMORIES, FULL·RANGE·
                ;          IF·THE·ANSWER·IS·'Y', TMT·BUILDS·A·COMPLETE·MEMORY·
                ;          TEST·CYCLING·TABLE, THEN·PROCEEDS·TO·5. (FOR·THE·
                ;          STRUCTURE·OF·THE·TABLE, SEE·BELOW·REF·TABLE·- CURRENT JUMP
                ;          TABLE).
                ;          IF·THE·ANSWER·IS·'N', PROCEED TO·2·FOR·MORE·SELECTIVE·PROMPTS.
                ;       2. PROMPT·FOR·MEMORIES TO·BE·TESTED·
                ;          TMT·SCANS THE·COMMAND·LINE·RESPONSE·AND SETS·A·FLAG·FOR·
                ;          EACH·MEMORY·MNEMONIC·IT·FINDS.
                ;       3. PROMPT·FOR·MEMORY·LIMITS·
                ;          IF·A·MEMORY·WAS·SELECTED·IN·2 (IE, ITS·FLAG·WAS·SET),
                ;          TMT·PROMPTS·FOR·NUMERICAL·VALUES·WHICH·DEFINE·THAT·
                ;          PORTION·OF·A·MEMORY ON·WHICH·THE·TESTS·ARE·TO·BE·RUN. TMT
                ;          PUTS·THE·MEMORY·LIMITS·FROM·THE·COMMAND LINE·RESPONSE·
                ;          INTO·A·TABLE.
                ;       4. PROMPT·FOR·TESTS·
                ;          TMT·SCANS THE·COMMAND·LINE·RESPONSE·FOR TEST·NUMBERS.
                ;          THEN, FOR EACH·MEMORY·WHOSE·FLAG·IS·UP, TMT·BUILDS·A·
                ;          ENTRY·IN·THE·CURRENT·JUMP·TABLE·(SEE·BELOW).
                ;       5. PROMPT·FOR·LOOP·ON·TEST·
                ;          TMT·SETS·A·FLAG·THAT·DETERMINES·WHETHER A·TEST·CYCLE·
                ;          WILL·BE·EXECUTED·ONCE·OR·EXECUTED·REPEATEDLY.
                ;       6. PROMPT·FOR·ERROR·OPTIONS·
                ;          TMT·SETS·FLAGS·WHICH·DETERMINE·WHAT·ACTION·WILL·BE·
                ;          TAKEN·IN·THE·EVENT·OF·AN·ERROR.
                ;       7. PUT·OUT·DIRECTIONS·FOR·STOPPING·TEST·
                ;          TMT·ALLOWS·THE·TEST CYCLE·TO·BE·INTERRUPTED·BY·
                ;          AN·UNSOLICTED·CHARACTER·INTERRUPT·FROM·THE·TERMINAL.
                ;       8. TEST·CYCLE·BEGINS·
                ;          NO·MORE·PROMPTING.
                ;
                ;
                ;       REFERENCE·TABLE·-·CURRENT·JUMP·TABLE·
                ;
                ;       THE·ACTUAL·EXECUTION·OF·A·TEST CYCLE·DEPENDS·UPON·THE·
                ;       CONTENTS·OF·THE·CURRENT·JUMP·TABLE. TMT·FILLS·IN·THIS·
                ;       TABLE·BY·MOVING·ENTRIES·FROM·THE REFERENCE TABLE·DEPENDING·
                ;       UPON·WHICH·MEMORIES·AND·WHICH·TESTS·WERE·SELECTED.
                ;
                ;       EACH·TEST·1 - 12·HAS·ITS·OWN·CONTROL·ROUTINE·IN·TMT.
                ;       THE·REFERENCE·TABLE·ENTRIES·ARE·THE·ADDRESSES·OF·THESE·
                ;       CONTROL·ROUTINES. FOR EACH TEST, THE REFERENCE TABLE
```

```
                ;       CONTAINS IN CONTIGUOUS POSITIONS THE CONTROL ROUTINE
                ;       ADDRESS REPLICATED A NUMBER OF TIMES, THE NUMBER OF
                ;       REPLICATIONS IS EQUAL TO THE NUMBER OF MEMORIES THAT
                ;       CAN BE TESTED (IE, 6, THE VALUE OF THE EQUATE 'MEM').
                ;       SO, FOR EXAMPLE, SINCE THERE ARE 6 MEMORIES, TEST 1'S
                ;       CONTROL ROUTINE ADDRESS WILL BE REPEATED IN THE REFERENCE
                ;       TABLE 6 TIMES:
                ;
                ;       .WORD   T1,T1,T1,T1,T1,T1
                ;
                ;       EACH ADDRESS HERE IS A PLACE-HOLDER FOR A MEMORY. THAT IS,
                ;       THE FIRST 'T1' (POSITION 0) IS A PLACE-HOLDER FOR FSA-A,
                ;       THE SECOND 'T1' (POSITION 1) IS A PLACE-HOLDER FOR FSA-B,
                ;       THE THIRD (POSITION 2) FOR FSA-C, THE FOURTH (POSITION 3)
                ;       FOR THE INPUT BUFFER, THE FIFTH (POSITION 4) FOR THE
                ;       OUTPUT BUFFER, AND THE SIXTH (POSITION 5) FOR THE BYTE
                ;       TRANSLATOR. THE IDEA HERE IS THAT RATHER THAN HAVING SEPARATE
                ;       CONTROL ROUTINES FOR EACH MEMORY FOR EACH TEST, TMT CAN
                ;       MAKE USE OF THE POSITIONS OF ADDRESSES IN THE TABLE.
                ;
                ;       THE FILLING OF THE CURRENT JUMP TABLE TAKES PLACE AS
                ;       FOLLOWS: A UNIQUE FLAG IS SET IN A FLAG WORD FOR EACH
                ;       MEMORY WHOSE MNEMONIC TMT ENCOUNTERS IN THE COMMAND
                ;       LINE RESPONSE THE THE PROMPT 'SELECT MEMORIES'. EACH
                ;       POSITION IN THE FLAG WORD (0 - 5) CORRESPONDS TO A
                ;       MEMORY PLACE HOLDER POSITION IN THE REF TABLE AND
                ;       CURRENT JUMP TABLE, EG. THE FLAG FOR FSA-A IS IN
                ;       POSITION 0 IN THE FLAG WORD AND THE PLACE-HOLDER
                ;       POSITION FOR FSA-A IS 0 (SEE ABOVE). THEN FOR EACH TEST
                ;       NUMBER TMT ENCOUNTERS IN THE COMMAND LINE RESPONSE TO
                ;       THE PROMPT 'SELECT TEST(S)', TMT MOVES THE ADDRESS OF THAT
                ;       TEST'S CONTROL ROUTINE FROM THE REF TABLE TO THE CURRENT
                ;       JUMP TABLE DEPENDING UPON THE MEMORY FLAG SETTINGS. IE, TMT
                ;       SCANS THE MEMORY FLAG WORD AND FOR EVERY BIT SET MOVES
                ;       AN ADDRESS FROM THE REF TABLE TO THE CURRENT JUMP TABLE.
                ;
                ;       EXAMPLE:
                ;       IF TMT ENCOUNTERS A '1' IN THE COMMAND LINE RESPONSE TO
                ;       'SELECT TEST(S)' AND THE FLAGS FOR FSA-A AND THE OUTPUT
                ;       BUFFER HAVE BEEN PREVIOUSLY SET, TMT WILL MOVE TEST 1'S
                ;       CONTROL ROUTINE ADDRESS FROM THE REF TABLE TO THE CURRENT
                ;       JUMP TABLE IN POSITIONS 0 AND 4 FOR TEST 1.
                ;
                ;       REF TABLE:
                ;       .WORD   T1,T1,T1,T1,T1,T1
                ;       .WORD   T2,T2,T2,T2,T2,T2
                ;
                ;       CURRENT JUMP TABLE:
                ;       .WORD   T1,0,0,0,T1,0
                ;       .WORD   0,0,0,0,0,0
                ;
                ;
                ;       MEMORY CYCLING
                ;
                ;       THE CYCLE CONTROL ROUTINE MAINTAINS A POINTER TO THE
                ;       CURRENT JUMP TABLE. THE CONTROL ROUTINE SCANS THE TABLE
                ;       UNTIL IT FINDS A NON-ZERO ENTRY. IT DERIVES THE MEMORY
```

```
                        ;       TO BE TESTED FROM THE PLACE-HOLDING CHARACTERISTICS OF
                        ;       THE TABLE:
                        ;
                        ;       1. GETS THE POSITION OF THE CURRENT NON-ZERO ENTRY
                        ;          RELATIVE TO THE BEGINNING OF THE TABLE.
                        ;       2. DIVIDES THIS NUMBER BY THE NUMBER OF MEMORIES. THE
                        ;          REMAINDER GIVES THE PLACE-HOLDER VALUE.
                        ;
                        ;       FOR EXAMPLE, IF 'INPUT BUFFER' AND 'TEST 2' HAVE BEEN
                        ;       SELECTED, THE CURRENT JUMP TABLE WILL CONTAIN THE
                        ;       FOLLOWING INFORMATION:
                        ;
                        ;       .WORD   0,0,0,0,0,0
                        ;       .WORD   0,0,0,T2,0,0
                        ;
                        ;       THE FIRST NON-ZERO ENTRY IN THE TABLE IS AT OFFSET 9.
                        ;       THE REMAINDER FROM THE DIVISION OF 9 BY 6 (6 = NUMBER
                        ;       OF MEMORIES) IS 3. THE VALUE 3 IS THE PLACE-HOLDER
                        ;       VALUE FOR THE INPUT BUFFER.
                        ;
                        ;       THE CONTROL ROUTINE PASSES CONTROL TO THE TEST CONTROL
                        ;       ROUTINE WHOSE ADDRESS IS THE CURRENT NON-ZERO ENTRY IN
                        ;       THE CURRENT JUMP TABLE. THE CYCLE CONTROL ROUTINE PASSES
                        ;       THE REMAINDER FROM THE ABOVE DIVISION IN R0. THE TEST
                        ;       CONTROL ROUTINE USES THE CONTENTS OF R0 AS AN INDEX
                        ;       INTO A TABLE OF TMT SUB-MODULE MEMORY TEST ADDRESSES.
                        ;       THE TEST CONTROL ROUTINE IN TURN PASSES CONTROL TO THE
                        ;       ROUTINE IN THE SUB-MODULE THAT WILL EXECUTE THE TEST ON
                        ;       THE CORRECT MEMORY.
                        ;
                        ;       ALL SUB-MODULE ROUTINES ARE LOCATED IN THE MODULE TTEST.
                        ;
                        ;       ALL OF THESE ROUTINES RETURN TO THE TEST CONTROL ROUTINE
                        ;       THAT CALLED THEM. THE TEST CONTROL ROUTINES RETURN TO THE
                        ;       CYCLING ROUTINE WHICH SCANS THE CURRENT JUMP TABLE FOR THE
                        ;       NEXT NON-ZERO ENTRY.
                        ;
                        ;
                        ;       EXIT FROM THE PROGRAM DEPENDS UPON THE STATUS OF THE CURRENT
                        ;       JUMP TABLE, LOOP OPTIONS, HALT OPTIONS, OR TERMINAL INPUT.
                        ;
                        ;       JUMP TABLE EMPTY        - EXIT.
                        ;       LOOP OPTION OFF         - EXECUTE ONE TEST CYCLE.
                        ;       LOOP COUNT              - EXECUTE A NUMBER OF CYCLES EQUAL TO
                        ;                                 THE LOOP COUNT.
                        ;       HALT OPTION ON          - HALT AFTER ONE ERROR.
                        ;       COUNT + 'H'             - PRINT A NUMBER OF MESSAGES EQUAL TO
                        ;                                 THE COUNT AND HALT.
                        ;
                        ;       WHILE THE TESTS ARE RUNNING, THE ENTERING FROM THE TERMINAL
                        ;       OF ANY CHARACTER OTHER THAN W, C, P, OR T (THESE HAVE SPECIAL
                        ;       MEANINGS - SEE THE ROUTINE 'AST') STOPS THE TESTS IMMEDIATELY.
                        ;
                        ;
                        ;       MEMORY TESTS:
                        ;
                        ;       TEST 01  WRITE MEMORY ADDRESS INTO ITSELF
```

```
172·                    ;       TEST·02·  WRITE·ZEROS··
173                     ;       TEST·03   WRITE·ONES·
174                     ;       TEST·04   WRITE·125252·
175                     ;       TEST·05   WRITE/READ·146314 AND·031463
176                     ;       TEST·06   MEMORY·CROSS-TALK·TEST·
177                     ;       TEST·07   WRITE·ADDRESS COMPLEMENT·INTO·ADDRESS·
178                     ;       TEST·08   WRITE·000377 AND·177400
179                     ;       TEST·09   SHIFT-BIT·TEST·
180                     ;       TEST·10   WRITE·USER·SUPPLIED·TEST·PATTERN·
181                     ;       TEST·11   BIT MARCH·TEST·
182·                    ;       TEST·12·  ADDRESSING·TEST·
183                     ;
184                     ;
185                     ;       ASSEMBLY:                       FROM·[5,3]·
186                     ;       MCR>MAC·TMT,LP=IM04,TMT·
187                     ;
188                     ;       TASK·BUILD:                     ;ON·TERM DETECTOR·PACK·
189                     ;       TMT,TMT=TMT,TTEST·
190                     ;       /·
191                     ;       PAR=PAR14K·
192·                    ;       TASK=...TMT·
193                     ;       ASG=TT0:1
194                     ;       ASG=TT0:2·
195                     ;       /·
196                     ;       //
```

```
 198                                    ;
 199                                    ;
 200                                    ;       LOCAL DATA AREAS
 201                                    ;
 202                                    ;
 203                                            .MCALL  QIOW$S,QIO$S,EXIT$S,ABRT$S,GCML$,GCMLB$,FSRSZ$,CLEF$S,ASTX$S
 204                                    ;
 205                                    ;
 206          000001                    LUN.TT  =       1               ;LUN FOR TT0
 207          000001                    EFN.1   =       1               ;EVENT FLAG FOR TT0
 208          000002                    CMILUN  =       2               ;LUN FOR GCML
 209                                    ;
 210                                    ;       SETTINGS OR FLAG WORD 'BASE'
 211                                    ;
 212          000001                    LOOP    =       1               ;LOOP FLAG
 213          000002                    TEST6   =       2               ;TEST 6 IN EFFECT
 214          000004                    TEST10  =       4               ;TEST 10 IN EFFECT
 215          000010                    ALLTST  =       10              ;ALL TESTS IN EFFECT
 216          000020                    HALT    =       20              ;HALT FLAG
 217          000040                    ERROR   =       40              ;ERROR HAS OCCURRED
 218                                    ;
 219                                    ;       SETTINGS FOR FLAG WORD 'SELECT'
 220                                    ;
 221          000001                    A       =       1               ;FSA - A
 222          000002                    B       =       2               ;FSA - B
 223          000004                    C       =       4               ;FSA - C
 224          000010                    I       =       10              ;INPUT BUFFER
 225          000020                    O       =       20              ;OUTPUT BUFFER
 226          000040                    T       =       40              ;BYTE TRANSLATOR
 227                                    ;
 228                                            .NLIST  BEX
 229 000000   131574   077434           MYSELF: .RAD50  /...TMT/        ;ABORT THIS TASK
 230 000004   300000                    ASTWRD: .WORD   0               ;RECEIVER FOR CHAR FROM TERMINAL
 231 000006                             STAT:   .BLKW   2               ;IO STATUS
 232 000012   000000                    ERWORD: .WORD   0               ;INFO/ERROR MESSAG INDEX
 233 000014   000000                    ERLIM:  .WORD   0               ;MAX ERROR MESSAGES TO PRINT
 234 000016   000000                    BINWD:  .WORD   0               ;VALUES CONVERTED FROM COMMAND LINE
 235 000020   000000                    UPPER:  .WORD   0               ;WORK FIELD FOR UPPER MEMORY LIMITS
 236 000022   000000                    LOWER:  .WORD   0               ;WORK FIELD FOR LOWER MEMORY LIMITS
 237 000024   000000                    BASE:   .WORD   0               ;GENERAL FLAG
 238 000026   000000                    MT10:   .WORD   0               ;USER TEST PATTERN
 239 000030   000000                    T6FACT: .WORD   0               ;ADDRESS SKIP FACTOR FOR TEST 6
 240 000032                             GCMBUF: .BLKW   41.             ;COMMAND LINE BUFFER
 241 000154   000000                    GCMLEN: .WORD   0               ;COMMAND LINE LENGTH
 242 000156   000000                    GCMPNT: .WORD   0               ;POINTER TO COMMAND LINE
 243 000160   000000                    PASSH:  .WORD   0               ;HIGH WORD OF PASS COUNT
 244 000162   000001                    PASS:   .WORD   1               ;LOW WORD OF PASS COUNT
 245 000164   000000                    LOOPCT: .WORD   0               ;LOOP COUNT
 246 000166   000000                    SELECT: .WORD   0               ;MEMORY SELECT FLAG WORD
 247 000170   000000                    MTPNT:  .WORD   0               ;CURRENT JUMP TABLE POINTER
 248 000172   000000                    MTCNT:  .WORD   0               ;CURRENT JUMP TABLE COUNT
 249 000174   000000                    SETTBL: .WORD   T$FSAA          ;SELECT FSA-A
 250 000176   000004                            .WORD   T$FSAB          ;SELECT FSA-B
 251 000200   000014                            .WORD   T$FSAC          ;SELECT FSA-C
 252                                    ;
 253                                    ;       TABLE USED IN TRANSLATING FROM ASCII DECIMAL TO BINARY
 254                                    ;
```

```
000202                               TROCT:
       000263'                       .       =       .+61
000263    001      002      003              .BYTE   1,2,3,4,5,6,7,8.,9.,10.,11.,12.
       000401'                       .       =       TROCT+177
                                     ;
                                     ;       STRING OF ALL TESTS (USED AS PSEUDO COMMAND LINE)
                                     ;
000401    061      040      062      ALLSTR: .ASCII  /1 2 3 4 5 6 7 8 9 11 12/
       000027                        STRLEN  =       .-ALLSTR
                                             .EVEN
                                     ;
                                     ;       VALID MEMORY MNEMONICS AND MEMORY SELECT FLAG SETTINGS
                                     ;
000430                               CTBL:
000430    101      115                       .ASCII  /AM/            ;FSA A
000432 000001                                .WORD   A
000434    102      115                       .ASCII  /BM/            ;FSA B
000436 000002                                .WORD   B
000440    103      115                       .ASCII  /CM/            ;FSA C
000442 000004                                .WORD   C
000444    111      102                       .ASCII  /IB/            ;INPUT BUFFER
000446 000010                                .WORD   I
000450    117      102                       .ASCII  /OB/            ;OUTPUT BUFFER
000452 000020                                .WORD   O
000454    102      124                       .ASCII  /BT/            ;BYTE TRANSLATOR
000456 000040                                .WORD   T
       000006                        CNUM    =       <.-CTBL/4>
                                     ;
                                     ;       MEMORY TEST CONTROL ROUTINE ADDRESSES (REFERENCE)
                                     ;
000460                               MTREF:
000460 006064' 006064' 006064'               .WORD   T1,T1,T1,T1,T1,T1
000474 006074' 006074' 006074'               .WORD   T2,T2,T2,T2,T2,T2
000510 006110' 006110' 006110'               .WORD   T3,T3,T3,T3,T3,T3
000524 006126' 006126' 006126'               .WORD   T4,T4,T4,T4,T4,T4
000540 006144' 006144' 006144'               .WORD   T5,T5,T5,T5,T5,T5
000554 006174' 006174' 006174'               .WORD   T6,T6,T6,T6,T6,T6
000570 006200' 006200' 006200'               .WORD   T7,T7,T7,T7,T7,T7
000604 006210' 006210' 006210'               .WORD   T8,T8,T8,T8,T8,T8
000620 006240' 006240' 006240'               .WORD   T9,T9,T9,T9,T9,T9
000634 006274' 006274' 006274'               .WORD   T10,T10,T10,T10,T10,T10
000650 006300' 006300' 006300'               .WORD   T11,T11,T11,T11,T11,T11
000664 006424' 006424' 006424'               .WORD   T12,T12,T12,T12,T12,T12
                                     ;
                                     ;       LOWER AND UPPER MEMORY LIMITS (REFERENCE)
                                     ;
000700                               LIMREF:
000700 001777                                .WORD   1777            ;FSA-A UPPER LIMIT
000702 000000                                .WORD   0
000704 001777                                .WORD   1777            ;FSA-B UPPER LIMIT
000706 000000                                .WORD   0
000710 001777                                .WORD   1777            ;FSA-C UPPER LIMIT
000712 000000                                .WORD   0
000714 163776                                .WORD   163776          ;INPUT BUFFER UPER LIMIT
000716 160000                                .WORD   160000
000720 003776                                .WORD   3776            ;OUTPUT BUFFER UPER LIMIT
000722 000000                                .WORD   0
```

```
                                                .WORD   255.            ;BYTE TRANSLATOR UPPER LIMIT
                                                .WORD   0
                                LIMNUM  =       <.-LIMREF>/2
                                ;
                                ;       MEMORY TEST ROUTINE ADDRESSES (CURRENT JUMP TABLE)
                                ;
                                TESTS   =       12.             ;NUMBER OF TESTS
                                MEM     =       6.              ;NUMBER OF MEMORIES
                                MTSUB:
                                                .BLKW   <TESTS*MEM>
                                ;
                                ;       LOWER AND UPPER MEMORY LIMITS (CURRENT TEST)
                                ;
                                CURLIM:
                                                .BLKW   LIMNUM
                                ;
                                ;
                                ;       SUB-MODULE MEMORY TEST ROUTINE ADDRESSES
                                ;
                                STADDR: .WORD   STUFF,STUFF,STUFF,STUFI,STUFO,STUFT
                                T1ADDR: .WORD   T1F,T1F,T1F,T1I,T1O,T1T
                                ;T6ADDR:        .WORD   T6F,T6F,T6F,T6I,T6O,T6T
                                T7ADDR: .WORD   T7F,T7F,T7F,T7I,T7O,T7T
                                TCDADD: .WORD   T12FD,T12FD,T12FD,T12ID,T12OD,T12TD
                                TCUADD: .WORD   T12FU,T12FU,T12FU,T12IU,T12OU,T12TU
                                ;
                                ;
                                ;       ERROR ROUTINE WORK AREAS
                                ;
                                CKDATA:: .WORD  0               ;TEST PATTERN
                                CK2::   .WORD   0               ;TEST 12 READ TEST PATTERN
                                CK3::   .WORD   0               ;TEST 12 WRITE TEST PATTERN
                                PREADD:: .WORD  0               ;CURRENT MEMORY ADDRESS
                                ERRADD:: .WORD  0               ;ADDRESS AT ERROR
                                ERRCT:: .WORD   0               ;NUMBER OF ERRORS
                                ERW1::  .WORD   0               ;ERRONEOUS RESULTS FROM MEMORY - 1
                                ERW2::  .WORD   0               ; - 2
                                ERW3::  .WORD   0               ; - 3
                                ;
    040     040     040 040     ASWRK:  .ASCII  /   /
    124     105     123 123     TMSG:   .ASCII  /TEST/
    120     101     123 123     PMSG:   .ASCII  /PASS/
    106     101     111 111     FMSG:   .ASCII  /FAILING MEMORY: /
    101     104     104 104     AMSG:   .ASCII  /ADDRESS: /
    105     130     120 120     EMSG:   .ASCII  /EXPECTED: /
    122     105     103 103     RMSG:   .ASCII  /RECEIVED: /
    052     040     052 052     UNMSG:  .ASCII  /* * */
                                ;
                                ;       MEMORY NAMES
                                ;
                                FTBL:
    106     123     101             .ASCII  /FSA-A/
    106     123     101             .ASCII  /FSA-B/
    106     123     101             .ASCII  /FSA-C/
    111     116     120             .ASCII  /INPUT/
    117     125     124             .ASCII  /OUTPUT/
    102     124     122             .ASCII  /BTRANS/
```

```
                                        ;
                                        ;
                                        ;       PRINT LINE
                                        ;
001463     015     012                  .BYTE   15,12                   ;PRECEDE PRINT LINE WITH CRLF
001465                                  PRINT:
                                        .NLIST  MEB
         000116                         .REPT   78.
                                        .BYTE   40
                                        .ENDR
                                        .LIST   MEB
                                        ;
                                        ;
                                        ;       TABLE OF MESSAGES
                                        ;
                                        ;
001603     000                          .BYTE   0
001604     015     012     015          .BYTE   15,12,15,12
001610     124     105     123          .ASCIZ  /TEST(S) ENDED/
001626     015     012     015          .BYTE   15,12,15,12
001632     124     105     123          .ASCIZ  /TEST(S) HALTED/
001651     015     012                  .BYTE   15,12
001653     105     116     124          .ASCIZ  /ENTER ANY CHARACTER TO STOP TEST(S)/
001717     015     012     015          .BYTE   15,12,15,12
001723     115     105     115          .ASCIZ  /MEMORY TEST(S) STARTED/
001752     015     012     015          .BYTE   15,12,15,12,15,12
001760     124     105     122          .ASCII  /TERM DETECTOR MEMORY TEST DIAGNOSTICS/
002025     015     012     000          .BYTE   15,12,0
002030     015     012                  .BYTE   15,12
002032     124     105     123          .ASCIZ  /TEST 6 INCOMPATABLE WITH MEMORY LIMITS. TEST DISCARDED./
002122     015     012                  .BYTE   15,12
002124     105     122     122          .ASCII  /ERROR: NO SELECTIONS, EXIT./
002157     015     012     000          .BYTE   15,12,0
002162     015     012                  .BYTE   15,12
002164     111     116     126          .ASCIZ  /INVALID ERROR OPTION/
002211     015     012                  .BYTE   15,12
002213     111     116     126          .ASCIZ  /INVALID LOOP OPTION/
002237     015     012                  .BYTE   15,12
002241     111     116     126          .ASCIZ  /INVALID TEST PATTERN/
002266     015     012                  .BYTE   15,12
002270     111     116     126          .ASCIZ  /INVALID TEST NUMBER/
002314     015     012                  .BYTE   15,12
002316     111     114     114          .ASCIZ  /ILLEGAL ODD ADDRESS/
002342     015     012                  .BYTE   15,12
002344     111     116     126          .ASCIZ  /INVALID UPPER MEMORY LIMITS/
002400     015     012                  .BYTE   15,12
002402     111     116     126          .ASCIZ  /INVALID LOWER MEMORY LIMITS/
002436     015     012                  .BYTE   15,12
002440     111     116     126          .ASCIZ  /INVALID MEMORY OR BUFFER MNEMONIC/
002502     015     012                  .BYTE   15,12
002504     111     116     126          .ASCIZ  /INVALID RESPONSE/
002525     015     012                  .BYTE   15,12
002527     105     116     124          .ASCIZ  /ENTER ERROR CONTROL/
002553     015     012                  .BYTE   15,12
002555     114     117     117          .ASCIZ  /LOOP ON TEST(S)?/
002576     015     012                  .BYTE   15,12
002600     105     116     124          .ASCIZ  /ENTER PATTERN FOR TEST 10/
```

```
    002632     015     012                      .BYTE   15.12
    002634     123     105     114              .ASCIZ  /SELECT TEST/
    002650     015     012                      .BYTE   15.12
    002652     105     116     124              .ASCIZ  /ENTER MEMORY LIMITS FOR BYTE TRANSLATOR/
    002722     015     012                      .BYTE   15.12
    002724     105     116     124              .ASCIZ  /ENTER MEMORY LIMITS FOR OUTPUT BUFFER/
    002772     015     012                      .BYTE   15.12
    002774     105     116     124              .ASCIZ  /ENTER MEMORY LIMITS FOR INPUT BUFFER/
    003041     015     012                      .BYTE   15.12
    003043     105     116     124              .ASCIZ  /ENTER MEMORY LIMITS FOR FSA-C/
    003101     015     012                      .BYTE   15.12
    003103     105     116     124              .ASCIZ  /ENTER MEMORY LIMITS FOR FSA-B/
    003141     015     012                      .BYTE   15.12
    003143     105     116     124              .ASCIZ  /ENTER MEMORY LIMITS FOR FSA-A/
    003201     015     012                      .BYTE   15.12
    003203     123     105     114              .ASCIZ  /SELECT MEMORY/
    003221     015     012                      .BYTE   15.12
    003223     101     114     114              .ASCIZ  /ALL TESTS, ALL MEMORIES, FULL RANGE ?/
    003271     377                      ASCIZ:  .BYTE   377
                                        ;
                                        ;
    003272                              PMSG2:
    003272     120     101     123              .ASCII  /PASS NUMBER /
            000014                      PM2LN   =       .-PMSG2
                                        ;
    003306                              ENDOF:
    003306     105     116     104              .ASCII  /END OF PASS /
            000014                      ENDLN   =       .-ENDOF
                                                .EVEN
                                                .LIST   BEX
                                                .N_IST  CND
                                        ;
                                        ;
                                        ;       COMMAND LINE MACRO
                                        ;
                                        ;
    003322                              GCMBLK: GCMLB$  2,,GCMBUF,CMILUN
    003322     002                              .BYTE   2
    003364     002                              .BYTE   CMILUN
    003370  003510'                             .WORD   65$
    003376     001                              .BYTE   1
    003341     000C                             .BYTE   FD:TTY!FD:REC!FD.CCL
    003462     000                              .BYTE   0
    003463     047                              .BYTE   GE:COM!GE.IND!GE.CLO!GE.SIZ
    003474     377     002                      .BYTE   -1,2
    003476  003550'                             .WORD   64$
    003500  000032'                             .WORD   GCMBUF
    003502     015     012                      .ASCII  <15><12>
    003504     040     040     040              .ASCII  /   >/
    003507     076
    003516  012321                              .RAD50  /CMI/
    003520  000000                              .WORD   0
    003522  000000                              .WORD   0
    003524  012314                              .RAD50  /CMD/
    003546  000120                              .WORD   80.
    003630                                      FSRSZ$  1
```

```
                                        ;
                                        ;
                                        ;       ENTER HERE.
                                        ;
                                        ;
                                        ;       PROMPT FOR ALL MEMORIES, ALL TESTS.
                                        ;
003630                                  START:
003630  004767  005240                          JSR     PC,OUT1         ;ISSUE INFORMATION MESSAGE.
003634  004767  005370                  ALL:    JSR     PC,ALLSEL       ;ISSUE PROMPT
003640  103003                                  BCC     1$              ;NEED A RESPONSE.
003642  004767  005302                          JSR     PC,ERR2         ;'INVALID RESPONSE'
003646  000772                                  BR      ALL             ;PROMPT AGAIN.
                                        ;
003650  004767  004420                  1$:     JSR     PC,FIND         ;LOCATE RESPONSE IN COMMAND LINE.
003654  103003                                  BCC     2$              ;OK, VALIDATE RESPONSE.
003656  004767  005266                          JSR     PC,ERR2
003662  000764                                  BR      ALL             ;PROMPT AGAIN.
                                        ;
                                        ;       PARSE RESPONSE.
                                        ;
003664  122711  000116                  2$:     CMPB    #'N,(R1)        ;N = NO.
003670  001435                                  BEQ     MTRTN           ;NOT ALL
003672  122711  000131                          CMPB    #'Y,(R1)        ;Y = YES
003676  001403                                  BEQ     MOVE            ;OK, SET UP FOR ALL
003700  304767  005244                          JSR     PC,ERR2         ;MUST BE Y OR N
003704  000753                                  BR      ALL             ;PROMPT AGAIN.
                                        ;
                                        ;       SET UP MEMORY LIMITS TABLE FOR CURRENT TEST,
                                        ;       COPY REFERENCE TABLE TO CURRENT TABLE (IE, TEST
                                        ;       MEMORIES OVER THEIR FULL RANGE).
                                        ;
003706  012700  000700'                 MOVE:   MOV     #LIMREF,R0      ;POINT TO REF TABLE.
003712  012701  001150'                         MOV     #CURLIM,R1      ;POINT TO CURRENT TABEL.
003716  012702  000014                          MOV     #LIMNUM,R2      ;NUMBER OF WORDS TO MOVE.
003722  012021                          1$:     MOV     (R0)+,(R1)+
003724  005302                                  DEC     R2
003726  001375                                  BNE     1$
                                        ;
                                        ;       MOVE ALL OF MEMORY TEST REFERENCE TABLE TO CURRENT JUMP
                                        ;       TABLE.
                                        ;
003730  012700  000460'                         MOV     #MTREF,R0       ;POINT TO REF TABLE.
003734  012701  000730'                         MOV     #MTSUB,R1       ;POINT TO CURRENT TABLE.
003740  012702  000110                          MOV     #<TESTS*MEM>,R2 ;NUMBER OF WORDS
003744  012021                          2$:     MOV     (R0)+,(R1)+
003746  005302                                  DEC     R2
003750  001375                                  BNE     2$
                                        ;
                                        ;       JUMP TO LOOP SELECTION
                                        ;
003752  052767  000010  174044                  BIS     #ALLTST,BASE    ;SET FLAG FOR ALL TESTS.
003760  000167  001046                          JMP     LPRMPT
```

```
519                                     ;
520                                     ;
521                                     ;       PROMPT FOR MEMORY SELECTION.
522                                     ;       IF RESPONSE IS <CR>, SELECT ALL MEMORIES.
523                                     ;
524                                     ;
525 003764                              MTRTN:
526 003764  005067  174176                      CLR     SELECT          ;CLEAR MEMORY SELECT FLAG.
527 003770  004767  005230                      JSR     PC,MEMSEL       ;PROMPT FOR MEMORY SELECTION
528 003774  004767  004274                      JSR     PC,FIND         ;FIND MEMORY MNEMONIC IN COMMAND LINE.
529 004000  103004                            BCC     MEMTOP          ;A MNEMONIC WAS FOUND.
530 004002  052767  000077  174156            BIS     #<A+B+C+I+O+T>,SELECT ;SET ALL MEMORIES SELECTED.
531 004010  000425                            BR      PLIM            ;NOW PROMPT FOR MEMORY LIMITS.
532                                     ;
533                                     ;       LOOP TO PROCESS MEMORY MNEMONICS (THERE MAY BE MORE
534                                     ;       THAN ONE IN THE COMMAND LINE). EG:
535                                     ;       >AM IB BT
536                                     ;
537 004012                              MEMTOP:
538 004012  022700  000002                      CMP     #2,R0           ;MNEMONIC LENGTH MUST BE 2.
539 004016  001403                            BEQ     2$              ;OK,CONTINUE.
540 004020  004767  005120                      JSR     PC,ERR3         ;INVALID MNEMONIC.
541 004024  000757                            BR      MTRTN           ;START OVER WITH PROMPT
542                                     ;
543                                     ;       MATCH MEMORY MNEMONIC AGAINST A TABLE OF VALID MNEMONICS
544                                     ;       AND THEIR ASSOCIATED MEMORY FLAG SETTINGS.
545                                     ;
546 004026  012700  000006              2$:     MOV     #CNUM,R0        ;NUMBER OF VALID MEMORY MNEMONICS
547 004032  012702  000430                      MOV     #CTBL,R2        ;POINT TO TABLE OF MNEMONICS
548 004036  004767  004140                      JSR     PC,SCAN         ;MATCH COMMAND LINE.
549 004042  103003                            BCC     3$              ;OK, CONTINUE.
550 004044  004767  005074                      JSR     PC,ERR3
551 004050  000745                            BR      MTRTN           ;START OVER.
552                                     ;
553                                     ;       SUBRTN SCAN SETS R1 -> FLAG.
554                                     ;       LOOK FOR NEXT MNEMONIC IN COMMAND LINE. IF THERE IS
555                                     ;       ONE, PROCESS IT. IF THERE IS NONE, FALL THROUGH TO
556                                     ;       PROMPT FOR MEMORY LIMITS.
557                                     ;
558 004052  051167  174110              3$:     BIS     (R1),SELECT     ;SET FLAG FOR MEMORY SELECTED.
559 004056  004767  004212                      JSR     PC,FIND         ;FIND NEXT MEMORY MNEMONIC IN COMMAND LINE.
560 004062  103353                            BCC     MEMTOP          ;SOMETHING MORE IS THERE.
```

```
                                ;
                                ;
                                ;       PROMPT FOR MEMORY LIMITS, READ AND VERFIY THEM.
                                ;
                                ;
                                ;       SCAN THE MEMORY SELECT FLAG WORD FROM POSITION 0 TO
                                ;       POSITION 5, FOR EVERY MEMORY WHOSE FLAG IS SET, CALL
                                ;       SUBRTN 'LIMITS'.
                                ;
004064                          PLIM:
004064  012700  000001          MOV     #A,R0           ;START TESTING WITH FSA-A
004070  012701  011220'         MOV     #PMPTA,R1       ;POINT TO FIRST PROMPT MESSAGE
004074  012702  000700'         MOV     #LIMREF,R2      ;POINT TO REFERENCE LIMITS TABLE
004100  012703  001150'         MOV     #CURLIM,R3      ;POINT TO CURRENT LIMITS TABLE
004104  012704  000006          MOV     #MEM,R4         ;LOOP COUNT = NUMBER OF MEMORIES
                                ;
004110  030067  174052          1$:     BIT     R0,SELECT       ;WAS MEMORY SELECTED
004114  001437                  BEQ     5$              ;NO, BUMP POINTERS
004116  010046                  2$:     MOV     R0,-(SP)        ;SAVE TEST BIT
004120  010146                  MOV     R1,-(SP)        ;ADDR OF PROMPT MESSAGE
004122  011246                  MOV     (R2),-(SP)      ;MOVE UPPER REF LIMITS
004124  016246  000002          MOV     2(R2),-(SP)     ;MOVE LOWER REF LIMITS
004130  004767  003646          JSR     PC,LIMITS
004134  012663  000002          MOV     (SP)+,2(R3)     ;MOVE IN CURRENT LOWER LIMITS
004140  012613                  MOV     (SP)+,(R3)      ;MOVE IN CURRENT UPPER LIMITS
004142  012601                  MOV     (SP)+,R1        ;RESTORE POINTER TO PROMPTS
004144  012600                  MOV     (SP)+,R0        ;RESTORE TEST BIT
                                ;
                                ;       INPUT BUFFER AND OUTPUT BUFFER INCREMENT THEIR
                                ;       ADDRESSES BY 2, DO NOT ALLOW AN ODD ADDRESS IN
                                ;       EITHER THEIR LOWER OR UPPER LIMITS.
                                ;
004146  032700  000010          BIT     #1,R0           ;CHECKING INPUT BUFFER
004152  001003                  BNE     3$              ;YES, DO NOT ALLOW ODD ADDRESS
004154  032700  000020          BIT     #0,R0           ;CHECKING OUTPUT BUFFER
004160  001415                  BEQ     5$              ;NO, SKIP ODD ADDRESS CHECK
004162  032763  000001  000002  3$:     BIT     #1,2(R3)        ;IS LOWER LIMIT ODD ?
004170  001403                  BEQ     4$              ;NO, CONTINUE
004172  004767  004732          JSR     PC,ERR6
004176  000747                  BR      2$              ;PROMPT AGAIN FOR LIMITS
004200  032713  000001          4$:     BIT     #1,(R3)         ;IS UPPER LIMIT ODD
004204  001403                  BEQ     5$              ;NO, CONTINUE
004206  004767  004716          JSR     PC,ERR6
004212  000741                  BR      2$              ;PROMPT AGAIN
                                ;
004214  005304                  5$:     DEC     R4              ;FINISHED ?
004216  001410                  BEQ     ENDLIM          ;YES
004220  006300                  ASL     R0              ;SHIFT TO TEST NEXT BIT
004222  162701  000004          SUB     #4,R1           ;BACK UP PROMPT ADDR POINTER
004226  062702  000004          ADD     #4,R2           ;BUMP REF POINTER
004232  062703  000004          ADD     #4,R3           ;BUMP CURRENT POINTER
004236  000724                  BR      1$              ;TEST NEXT
```

```
                                        ;
                                        ;
                                        ;       SELECT TEST(S)
                                        ;
                                        ;
                                        ;       PROMPT FOR TEST NUMBERS.  IF THE RESPONSE IS <CR>
                                        ;       (CARRAIGE RETURN ONLY), MOVE A PSEUDO COMMAND LINE
                                        ;       INTO THE COMMAND LINE BUFFER.  THIS PSEUDO LINE
                                        ;       CONSISTS OF THE TEST NUMBERS FOR ALL TESTS EXCEPT
                                        ;       TEST 10 (USER PATTERN).  PROCEED TO PROCESS THIS
                                        ;       LINE AS THOUGH IT WAS ENTERED FROM THE TERMINAL.
                                        ;
004240                                  ENDLIM:
004240  004767  004724                          JSR     PC,SELTST       ;PROMPT TO SELECT A TEST
004244  004767  004024                          JSR     PC,FIND         ;SCAN COMMAND LINE FOR A TEST NUMBER
004250  103021                          BCC     SELOOP          ;FOUND A TEST NUMBER
004252  012700  000401'                  MOV     #ALLSTR,R0      ;POINT TO A STRING WITH ALL TESTS
004256  012701  000032'                  MOV     #GCMBUF,R1      ;POINT TO COMMAND BUFFER
004262  012702  000027                  MOV     #STRLEN,R2      ;GET LENGTH OF STRING
004266  112021                  1$:     MOVB    (R0)+,(R1)+     ;SIMULATE ASKING FOR ALL TESTS
004270  005302                          DEC     R2
004272  001375                          BNE     1$
004274  012767  000027  173652          MOV     #STRLEN,GCMLEN  ;MOVE LENGTH OF SIMULATED STRING
004302  012767  000032' 173646          MOV     #GCMBUF,GCMPNT  ;INIT POINTER TO COMMAND LINE
004310  004767  003760                  JSR     PC,FIND         ;LOCATE FIRST TEST NUMBER IN PSEUDO LINE
                                        ;
                                        ;       PROCESS ONE TEST NUMBER AT A TIME.  FIRST VALIDATE
                                        ;       THE NUMBER.
                                        ;
004314                                  SELOOP:
004314  022700  000002                  CMP     #2,R0           ;TEST NUMBERS ARE 1 BYTE EACH
004320  002416                          BLT     SERR            ;NO GOOD
004322  003007                          BGT     1$              ;SINGLE DIGIT, PROCESS IT
004324  122721  000061                  CMPB    #'1,(R1)+       ;TENS DIGIT?
004330  001012                          BNE     SERR            ;MUST BE 1
004332  111103                          MOVB    (R1),R3         ;LOAD 1'S DIGIT
004334  062703  000012                  ADD     #10.,R3         ;ADD VALUE OF TENS DIGIT
004340  000401                          BR      TRT             ;AND CONTINUE
                                        ;
                                        ;       TRANSLATE SINGLE DIGIT FROM ASCII DECIMAL TO BINARY.
                                        ;       IF A ZERO VALUE IS RETURNED FROM THE TRANSLATION,
                                        ;       THE ASCII CHARACTER IS INVALID.
                                        ;
004342  111103                  1$:     MOVB    (R1),R3         ;LOAD TEST NUMBER (ASCII)
004344  012704  000202'          TRT:    MOV     #TROCT,R4       ;POINT TO TRANSLATE TABLE
004350  060304                          ADD     R3,R4           ;ADD VALUE OF ASCII NUMBER
004352  111403                          MOVB    (R4),R3         ;MOVE BINARY VALUE FROM TABLE
004354  001012                          BNE     ZREL            ;SOMETHING WAS THERE
                                        ;
                                        ;       INVALID CHARACTER.  PUT OUT MESSAGE, CLEAR CURRENT JUMP
                                        ;       TABLE AND GO BACK TO PROMPT AGAIN.
                                        ;
004356  004767  004542          SERR:   JSR     PC,ERR7
004362  012700  000730'                  MOV     #MTSUB,R0       ;POINT TO JUMP TABLE
004366  012701  000110                  MOV     #<TESTS*MEM>,R1 ;LOAD NUMBER OF WORDS IN TABLE
004372  005020                  3$:     CLR     (R0)+           ;RESET TABLE
004374  005301                          DEC     R1
```

```
672 004376  001375                        BNE     3$
673 004400  000717                        BR      ENDLIM          ;TRY AGAIN
674                                 ;
675                                 ;       IF TEST NUMBER = 6 OR 10, SET A FLAG FOR
676                                 ;       LATER ACTION.
677                                 ;
678 004402                          ZREL:
679 004402  122703  000012              CMPB    #10.,R3         ;TEST 10 (USER TEST PATTERN)
680 004406  001003                      BNE     1$              ;NO, DO NOT PROMPT
681 004410  052767  000004  173406      BIS     #TEST10,BASE    ;PROMPT FOR TEST PATTERN (LATER)
682 004416  122703  000006          1$: CMPB    #6,R3           ;TEST 6 (CROSS TALK)
683 004422  001003                      BNE     2$              ;NO, DO NOT PROMPT
684 004424  052767  000002  173372      BIS     #TEST6,BASE     ;SET FLAG FOR LIMITS CHECK LATER
685                                 ;
686                                 ;
687                                 ;       MAKE TEST NUMBER ZERO-RELATIVE, MULTIPLY THE ZERO-
688                                 ;       RELATIVE TEST NUMBER BY THE NUMBER OF MEMORIES X 2
689                                 ;       TO GET A BYTE OFFSET INTO THE REFERENCE TABLE AND
690                                 ;       CURRENT JUMP TABLE. FOR EXAMPLE, IF THE ASCII TEST NUMBER
691                                 ;       WAS 2, THE ZERO-RELATIVE NUMBER IS 1. THIS NUMBER IS
692                                 ;       MULTIPLIED BY 12 TO GET A BYTE OFFSET = 12.
693                                 ;
694                                 ;       ADD THE PRODUCT TO THE START ADDRESS OF THE CURRENT JUMP
695                                 ;       TABLE AND PUT THE RESULT IN R0. ADD THE SAME PRODUCT TO
696                                 ;       THE START ADDRESS OF THE REF TABLE AND PUT THE RESULT IN
697                                 ;       R1. THE RESULTS ARE:
698                                 ;
699                                 ;       CURRENT JUMP TABLE (ASSUMING TEST 2 SELECTED)
700                                 ;       .WORD   0,0,0,0,0,0
701                                 ;       .WORD   0,0,0,0,0,0
702                                 ;
703                                 ;       REFERENCE TABLE
704                                 ;       .WORD   T1,T1,T1,T1,T1,T1
705                                 ;       .WORD   T2,T2,T2,T2,T2,T2
706                                 ;
707                                 ;       R0 -> FIRST 0 IN THE SECOND LINE FOLLOWING 'CURRENT JUMP TABLE'
708                                 ;       R1 -> FIRST T2 IN THE REFERENCE TABLE
709                                 ;
710                                 ;
711 004432  005303                  2$:     DEC     R3              ;MAKE NUMBER ZERO RELATIVE
712 004434  010301                      MOV     R3,R1           ;SYSTEM SUBRTN EXPECTS MULTIPLICAND IN R1
713 004436  012700  000014              MOV     #<MEM*2>,R0     ;AND MULTIPLIER IN R0
714 004442  004767  000000G             JSR     PC,$MUL
715 004446  010103                      MOV     R1,R3           ;MOVE PRODUCT TO R3
716 004450  012700  000730'             MOV     #MTSUB,R0       ;POINT TO JUMP TABLE
717 004454  060300                      ADD     R3,R0           ;ADD OFFSET
718 004456  012701  000460'             MOV     #MTREF,R1       ;TABLE WITH ALL SUBRTN ADDRESSES (REFERENCE)
719 004462  060301                      ADD     R3,R1           ;ADD OFFSET
720                                 ;
721                                 ;       DEPENDING UPON WHICH MEMORIES HAVE BEEN SELECTED FOR TESTING,
722                                 ;       MOVE ADDRESSES OF ROUTINES THAT GOVERN TESTS FROM THE REFERENCE
723                                 ;       TABLE TO THE JUMP TABLE (MTSUB). START TESTING THE MEMORY
724                                 ;       SELECT FLAG AT POSITION 0 (FSA-A).
725                                 ;
726                                 ;       PROCEEDING WITH THE ABOVE EXAMPLE ASSUMING IN ADDITION THAT
727                                 ;       FSA-B WAS THE MEMORY SELECTED, THIS ROUTINE WOULD FILL
728                                 ;       THE CURRENT JUMP TABLE IN THE FOLLOWING MANNER:
```

```
                                        ;
                                        ;       .WORD   0,0,0,0,0,0
                                        ;       .WORD   0,T2,0,0,0,0
                                        ;
004464  012702  000001                  MOV     #A,R2           ;START WITH FSA-A
004470  012703  000006                  MOV     #MEM,R3         ;TOTAL NUMBER OF MEMORIES
004474  030267  173466          3$:     BIT     R2,SELECT       ;WAS MEMORY SELECTED
004500  001401                          BEQ     4$
004502  011110                          MOV     (R1),(R0)       ;MOVE FROM REF TO CURRENT
004504  022120          4$:             CMP     (R1)+,(R0)+     ;BUMP POINTERS
004506  006302                          ASL     R2              ;SHIFT TO TEST NEXT BIT
004510  005303                          DEC     R3              ;FINISHED ?
004512  001370                          BNE     3$
                                        ;
                                        ;       TRANSFERS BETWEEN REF TABLE AND CURRENT JUMP TABLE ARE
                                        ;       COMPLETE FOR ONE TEST NUMBER. NOW SCAN THE COMMAND LINE
                                        ;       FOR THE NEXT TEST NUMBER.
                                        ;
004514  004767  003554                  JSR     PC,FIND         ;FIND NEXT TEST REQUEST IN COMMAND LINE
004520  103275                          BCC     SELOOP          ;PROCESS IT
```

```
                                        ;
                                        ;
                                        ;       FINISHED WITH COMMAND LINE FOR TESTS,
                                        ;       PROMPT FOR TEST PATTERN (IF TEST 10 WAS SELECTED).
                                        ;
                                        ;
004522  032767  000004  173274          PMPT10: BIT     #TEST10.BASE    ;WAS TEST 10 SELECTED
004530  001421                                  BEQ     T6CHK           ;NO, SKIP ALL THIS
004532  004767  004426                          JSR     PC,USRTST       ;PROMPT
004536  004767  003532                          JSR     PC,FIND         ;FIND A NON-BLANK IN COMMAND LINE
004542  103003                                  BCC     1$              ;OK, CONTINUE
004544  004767  004350                          JSR     PC,ERR0
004550  000770                                  BR      PMPT10          ;TRY AGAIN
004552  004767  003634                  1$:     JSR     PC,PACK         ;CONVERT TEST PATTERN TO BINARY
004556  103003                                  BCC     2$              ;CONVERSION OK
004560  004767  004334                          JSR     PC,ERR0
004564  000762                                  BR      PMPT10
004566  016767  173224  173232          2$:     MOV     BINWD,MT10      ;PUT PATTERN IN A SAFE PLACE
                                        ;
                                        ;
                                        ;       CHECK FLAG TO SEE WHETHER TEST 6 WAS SELECTED, IF
                                        ;       IT WAS, SET UP A POINTER TO THE CURRENT JUMP TABLE
                                        ;       ENTRIES FOR TEST 6.
                                        ;
                                        ;
004574                                  T6CHK:
004574  032767  000002  173222                  BIT     #TEST6.BASE     ;WAS TEST 6 SELECTED
004602  001476                                  BEQ     CHECK0          ;NO, SKIP AROUND
004604  005001                                  CLR     R1              ;START PLACE-HOLDER VALUE = 0
004606  012700  001024                          MOV     #MTSUB+<5*<MEM*2>>,R0 ;POINT TO TEST 6 ADDRESSES
004612  012702  000006                          MOV     #MEM,R2         ;LOOP COUNT = NUMBER OF MEMORIES
                                        ;
                                        ;       FOR EACH MEMORY TO COME UNDER TEST 6, CHECK WHETHER
                                        ;       THE MEMORY LIMITS ARE COMPATABLE WITH THE TEST; TEST 6
                                        ;       REQUIRES AT LEAST THREE MEMORY LOCATIONS IN ORDER TO
                                        ;       WORK CORRECTLY.
                                        ;
                                        ;
                                        ; **********************
                                        ;
                                        ;       HOW TEST 6 WORKS, RATIONALE BEHIND 3 LOCATION RULE.
                                        ;
                                        ;       TEST 6 CLEARS MEMORY FROM THE LOWER TO THE UPPER LIMIT,
                                        ;       IT THEN WRITES ALL 1'S IN THE FIRST LOCATION AND EVERY
                                        ;       OTHER LOCATION TO THE UPPER LIMIT, IT THEN READS ZEROS
                                        ;       FROM THE LOCATIONS INTO WHICH IT DID NOT WRITE 1'S, IT
                                        ;       THEN BUMPS THE LOWER LIMIT BY ONE MEMORY INCREMENT (VALUE
                                        ;       VARIES DEPENDING UPON THE MEMORY), IT CLEARS MEMORY UP TO
                                        ;       THE UPPER LIMIT, IT WRITES 1'S INTO THE NEW LOWER LIMIT
                                        ;       AND EVERY OTHER LOCATION TO THE UPPER LIMIT, IT READS
                                        ;       ZEROS FROM THE LOCATIONS INTO WHICH IT DID NOT WRITE
                                        ;       1'S, THREE MEMORY LOCATIONS ARE THE MINIMUM ON WHICH
                                        ;       TEST 6 CAN WORK:
                                        ;
                                        ;       WRITE 1'S INTO LOCATIONS 0 AND 2
                                        ;       READ ZEROS FROM LOCATION 1
                                        ;       177777
```

```
                                ;       000000
                                ;       177777
                                ;
                                ;       BUMP LOWER LIMIT TO 1
                                ;       CLEAR LOCATIONS 1 AND 2
                                ;       WRITE 1'S INTO LOCATION 1
                                ;       READ ZEROS FROM LOCATION 2
                                ;       177777
                                ;       177777
                                ;       000000
                                ;
                                ; **********
                                ;
                                ;
                                ;       CHECK EACH NON-ZERO TEST 6 ENTRY IN THE CURRENT JUMP TABLE,
                                ;       FOR EACH NON-ZERO ENTRY, SET UP IN R4 THE MEMORY INCREMENT
                                ;       VALUE X 2 (IE, THE MINIMUM SPREAD FOR TEST 6), BASIC VALUES:
                                ;
                                ;       FSA-A   1
                                ;       FSA-B   1
                                ;       FSA-C   1
                                ;       INPUT   2
                                ;       OUTPUT  2
                                ;       BTRANS  1
                                ;
004616  005710                  CHECK:  TST     (R0)            ;TEST 6 SELECTED FOR THIS MEMORY?
004620  001414                          BEQ     2$              ;NO, SKIP CHECK
004622  022701  000005                  CMP     #5,R1           ;BYTE TRANSLATOR
004626  001003                          BNE     1$
004630  012704  000002                  MOV     #2,R4           ;VALUE X 2
004634  000416                          BR      4$
004636  022701  000004          1$:     CMP     #4,R1           ;OUTPUT BUFFER
004642  001003                          BNE     2$
004644  012704  000004                  MOV     #4,R4           ;VALUE X 2
004650  000410                          BR      4$
004652  022701  000003          2$:     CMP     #3,R1           ;INPUT BUFFER
004656  001003                          BNE     3$
004660  012704  000004                  MOV     #4,R4           ;VALUE X 2
004664  000402                          BR      4$
004666  012704  000002          3$:     MOV     #2,R4           ;VALUE X 2 FOR FSA'S
                                ;
                                ;       USE THE PLACE-HOLDER VALUE AS AN INDEX INTO THE CURRENT
                                ;       LIMITS TABLE (AFTER SHIFTING FOR DOUBLE-WORD OFFSET), THEN
                                ;       CHECK THE RELATIONSHIP BETWEEN LOWER AND UPPER LIMITS.
                                ;
004672  010103                  4$:     MOV     R1,R3           ;SHIFT IN ANOTHER REG
004674  006303                          ASL     R3
004676  006303                          ASL     R3              ;SHIFT FOR DOUBLE WORD OFFSET
004700  066304  001152'                  ADD     CURLIM+2(R3),R4 ;GET LOWER LIMIT + MINIMUM INCR
004704  026304  001150'                  CMP     CURLIM(R3),R4   ;IS UPPER LIMIT OK FOR TEST 6
004710  103026                          BHIS    6$              ;YES, CONTINUE
                                ;
                                ;       FAILED CHECK, CLEAR THE ENTRY IN THE CURRENT JUMP TABLE
                                ;       AND REPORT TO THE CONSOLE, R1 = PLACE-HOLDER VALUE, THIS
                                ;       VALUE IS USED AS AN INDEX INTO A TABLE OF MEMORY NAMES
                                ;       (AFTER MULTIPLYING THE VALUE BY 6, THE LENGTH OF EACH
                                ;       NAME)
```

```
                                        ;
                                        ;       PREPARE TO CHECK NEXT TEST 6 ENTRY
                                        ;
004766  005201                  6$:    INC     R1              ;BUMP MEMORY PLACE-HOLDER VALUE
004770  062700  000002                 ADD     #2,R0           ;POINT TO NEXT TEST 6 ADDRESS
004774  005302                         DEC     R2              ;SUB FROM LOOP COUNT
004776  001307                         BNE     CHECK
                                        ;
                                        ;
                                        ;       MAKE SURE THAT THE 'CURRENT JUMP TABLE' IS NOT EMPTY
                                        ;
                                        ;
005000                          CHECK0:
005000  012700  000730'                 MOV     #MTSUB,R0       ;POINT TO CURRENT TABLE
005004  012701  000110                 MOV     #<MEM*TESTS>,R1 ;NUMBER OF TABLE ENTRIES
005010  005720                 1$:     TST     (R0)+           ;IS A TABLE ENTRY PRESENT
005012  001007                         BNE     LPRMPT          ;YES, EXIT THIS ROUTINE
005014  005301                         DEC     R1              ;SUB FROM ROUTINE COUNT
005016  001374                         BNE     1$              ;TRY NEXT POSITION
005020  004767  004060                 JSR     PC,ERR11        ;EXECUTION IMPOSSIBLE
                                        ;
005024                                  EXIT$S
005024  012746                         MOV     (PC)+,-(SP)
005026     063     001                 .BYTE   51.,1
005030  104377                         EMT     ^O<377>
```

```
                                        ;
                                        ;
                                        ;       PROMPT FOR LOOP ON TEST
                                        ;
                                        ;
                                        ;       RESPONSES:
                                        ;       <CR>            - CARRAIGE RETURN, LOOP ON TESTS.
                                        ;       Y               - YES, LOOP ON TESTS.
                                        ;       N               - NO, ONE MEMORY TEST CYCLE ONLY.
                                        ;       NUMERIC VALUE   - NUMBER OF CYCLES TO EXECUTE.
                                        ;
005032                                  LPRMPT:
005032  004767  004122                          JSR     PC,LPTST        ;PROMPT
005036  004767  003232                          JSR     PC,FIND         ;FIND RESPONSE IN COMMAND LINE
005042  103004                                  BCC     1$              ;OK, RESPONSE FOUND
005044  052767  000001  172752                  BIS     #LOOP,BASE      ;CR RESPONSE MEANS LOOP
005052  000442                                  BR      ERPRMT          ;AND CONTINUE
                                        ;
005054  122711  000131                  1$:     CMPB    #'Y,(R1)        ;YES - LOOP ON TESTS
005060  001004                                  BNE     2$              ;TRY 'N'
005062  052767  000001  172734                  BIS     #LOOP,BASE      ;SET FLAG FOR LOOP
005070  000433                                  BR      ERPRMT          ;PROMPT FOR ERROR OPTIONS
005072  122711  000116                  2$:     CMPB    #'N,(R1)        ;NO - DO NOT LOOP ON TESTS
005076  001004                                  BNE     3$              ;NO, TEST FOR LOOP COUNT
005100  042767  000001  172716                  BIC     #LOOP,BASE      ;CLEAR LOOP FLAG
005106  000424                                  BR      ERPRMT
                                        ;
                                        ;       ASSUME THAT THERE IS AN ASCII DECIMAL VALUE IN THE COMMAND
                                        ;       LINE, CONVERT IT TO BINARY AND STORE.
                                        ;
005110  060100                          3$:     ADD     R1,R0           ;POINT 1 PAST STRING
005112  005200                                  INC     R0              ;BUMP FOR STUPID SYSTEM SUBRTN
005114  010046                                  MOV     R0,-(SP)        ;SAVE FOR LATER COMPARISON
005116  010100                                  MOV     R1,R0           ;MOVE POINTER TO R0 FOR SYSTEM SUBRTN
005120  004767  000000G                        JSR     PC,$CDTB        ;CONVERT DECIMAL TO BINARY
005124  020026                                  CMP     R0,(SP)+        ;WHOLE STRING CONVERTED
005126  001403                                  BEQ     4$              ;YES, CONTINUE
005130  004767  003760                        JSR     PC,ERR9
005134  000736                                  BR      LPRMPT          ;PROMPT AGAIN
005136  010167  173022                  4$:     MOV     R1,LOOPCT       ;SAVE LOOP COUNT
005142  001003                                  BNE     5$
005144  004767  003744                        JSR     PC,ERR9
005150  000730                                  BR      LPRMPT
005152  052767  000001  172644          5$:     BIS     #LOOP,BASE      ;SET LOOP FLAG
```

```
                                ;
                                ;
                                ;       PROMPT FOR ERROR OPTIONS
                                ;
                                ;
                                ;       RESPONSES:
                                ;       <CR>            - CARRAIGE RETURN. DISPLAY ALL ERRORS.
                                ;       H               - HALT. STOP TESTS AFTER 1ST ERROR.
                                ;       NUMERIC VALUE   - PRINT ONLY THIS NUMBER OF ERROR MESSAGES
                                ;                         BUT CONTINUE TESTS.
                                ;       N. VALUE + H    - PRINT THIS NUMBER OF ERROR MESSAGES AND HALT.
                                ;
005160                          ERPRMT:
005160  004767  003770                  JSR     PC,EROPT        ;PROMPT FOR OPTIONS
005164  004767  003104                  JSR     PC,FIND         ;FIND RESPONSE
005170  103445                          BCS     MTSET           ;<CR> - DEFAULT TO 'DISPLAY'
                                ;
005172  122711  000110                  CMPB    #'H,(R1)        ;HALT AFTER 1 ERROR
005176  001007                          BNE     1$              ;NO
005200  052767  000020  172616          BIS     #HALT,BASE      ;SET FLAG FOR HALT
005206  012767  000002  172600          MOV     #2,ERLIM        ;PRINT ONLY 1 ERROR MESSAGE
005214  000433                          BR      MTSET
                                ;
                                ;       ASSUME AN ASCII DECIMAL VALUE IN THE COMMAND LINE, CONVERT
                                ;       AND STORE.
                                ;
005216  060100                  1$:     ADD     R1,R0           ;POINT PAST STRING IN COMMAND LINE
005220  005200                          INC     R0              ;BUMP FOR STUPID SYSTEM SUBRTN
005222  010046                          MOV     R0,-(SP)        ;SAVE FOR LATER COMPARISON
005224  010100                          MOV     R1,R0           ;PREPARE TO CONVERT
005226  004767  000000G                 JSR     PC,$CDTB
005232  020026                          CMP     R0,(SP)+        ;ENTIRE STRING CONVERTED
005234  001403                          BEQ     2$              ;YES
005236  004767  003646                  JSR     PC,ERR10
005242  000746                          BR      ERPRMT          ;TRY AGAIN
                                ;
                                ;       LOOK FOR AND 'H' AFTER THE ASCII OCTAL VALUE.
                                ;
005244  010167  172544          2$:     MOV     R1,ERLIM        ;NUMBER OF ERROR MSGS TO PRINT
005250  005267  172540                  INC     ERLIM           ;ADJUST FOR PRE-DECREMENT
005254  004767  003014                  JSR     PC,FIND         ;SCAN COMMAND LINE
005260  103411                          BCS     MTSET           ;NOTHING ELSE THERE
005262  122711  000110                  CMPB    #'H,(R1)        ;HALT AFTER MESSAGE COUNT EXHAUSTED
005266  001403                          BEQ     4$              ;YES
005270  004767  003614                  JSR     PC,ERR10        ;BAD OPTION
005274  000731                          BR      ERPRMT          ;TRY AGAIN
                                ;
005276  052767  000020  172520  4$:     BIS     #HALT,BASE      ;SET HALT FLAG
```

```
 997                                    ;
 998                                    ;
 999                                    ;       PREPARE TO ENTER MAIN LOOP
1000                                    ;
1001                                    ;
1002                                    ;       INITIALIZE CURRENT JUMP TABLE POINTER AND COUNTER
1003                                    ;       ADJUST FOR PRE-DCREMENT OF COUNT, PRE-INCREMENT OF POINTER
1004                                    ;
1005 005304                            MTSET:
1006 005304  012767  000726' 172656            MOV     #MTSUB-2,MTPNT  ;SET UP POINTER TO JUMP TABLE
1007 005312  012767  000111  172652            MOV     #<TESTS*MEM>+1,MTCNT ;AND NUMBER OF JUMP TABLE ENTRIES
1008 005320  012767  000000  176376            MOV     #0,TD$SW        ;RESET TD
1009                                    ;
1010                                    ;       THE STOP MESSAGE SAYS 'ENTER ANY CHARACTER TO STOP TEST(S)',
1011                                    ;       ACTUALLY THE CHARACTERS W, P, C, AND T ARE SPECIAL IN THAT
1012                                    ;       THEY ALLOW THE RUN TO CONTINUE AFTER THE PRINTING OUT OF
1013                                    ;       STATUS INFORMATION, SEE THE ROUTINE 'AST',
1014                                    ;
1015 005326  004767  003536                    JSR     PC,BEGTST       ;PUT OUT START TEST MESSAGE
1016 005332  004767  003526                    JSR     PC,STOP         ;GIVE DIRECTIONS FOR STOPPING TEST
1017 005336                                    QIO$S   #IO.ATA,#LUN.TT,,,,,<#AST>
     005336  005046                            CLR     -(SP)
     005340  005046                            CLR     -(SP)
     005342  005046                            CLR     -(SP)
     005344  005046                            CLR     -(SP)
     005346  005046                            CLR     -(SP)
     005350  012746  007206'                   MOV     #AST,-(SP)
     005354  005046                            CLR     -(SP)
     005356  005046                            CLR     -(SP)
     005360  005046                            CLR     -(SP)
     005362  012746  000001                    MOV     #LUN.TT,-(SP)
     005366  012746  000000G                   MOV     #IO.ATA,-(SP)
     005372  012746                            MOV     (PC)+,-(SP)
     005374     001     014                    .BYTE   1,12.
     005376  104377                            EMT     ^O<377>
```

```
1019                                    ;
1020                                    ;
1021                                    ;       MAIN LOOP OF MEMORY TEST FUNCTION
1022                                    ;
1023                                    ;
1024                                    ;       CHECK FOR THE HALT OPTION.  IF THE OPTION IS
1025                                    ;       ON, CHECK TO SEE WHETHER ANY ERRORS HAVE OCCURRED.
1026                                    ;       IF THEY HAVE (AND THE HALT COUNT IS EXHAUSTED),
1027                                    ;       TERMINATE THE TESTS.
1028                                    ;
1029 005400                            MTMAIN:
1030 005400  032767  000020  172416            BIT     #HALT,BASE      ;HALT AFTER ERROR
1031 005406  001413                            BEQ     1$              ;NO
1032 005410  032767  000040  172406            BIT     #ERROR,BASE     ;HAS AN ERROR OCCURRED
1033 005416  001407                            BEQ     1$              ;NO
1034 005420  026727  172370  000001            CMP     ERLIM,#1        ;IS REMAINING PRINTOUT COUNT 1 OR LESS
1035 005426  003003                            BGT     1$              ;NO, CONTINUE
1036 005430  004767  003424                    JSR     PC,HLTTST       ;PUT OUT HALT MESSAGE
1037 005434  000464                            BR      8$              ;AND EXIT
1038                                    ;
1039                                    ;       FIND A NON-ZERO ENTRY IN THE 'CURRENT JUMP TABLE'.
1040                                    ;       (A NON-ZERO ENTRY IS THE ADDRESS OF A TEST CONTROL
1041                                    ;       ROUTINE).  IF NO NON-ZERO ENTRIES ARE FOUND BEFORE
1042                                    ;       THE END OF THE TABLE IS REACHED, THEN ONE MEMORY TEST
1043                                    ;       CYCLE OR 'PASS' IS COMPLETE.
1044                                    ;
1045 005436  016701  172526            1$:     MOV     MTPNT,R1        ;POINT TO JUMP TABLE
1046 005442  005367  172524            2$:     DEC     MTCNT           ;FIRST SUB FROM #.RTNS LEFT
1047 005446  001406                            BEQ     3$              ;ALL DONE, TEST LOOP FLAG
1048 005450  005721                            TST     (R1)+           ;ADVANCE POINTER
1049 005452  005711                            TST     (R1)            ;IS THERE AN ADDRESS IN THE TABLE
1050 005454  001772                            BEQ     2$              ;NO, BUMP TO NEXT
1051 005456  010167  172506                    MOV     R1,MTPNT        ;SAVE JUMP TABLE POINTER
1052 005462  000454                            BR      JMPMT           ;AND JUMP TO ROUTINE
1053                                    ;
1054                                    ;
1055                                    ;       PASS FINISHED
1056                                    ;
1057                                    ;
1058                                    ;       IF THE LOOP FLAG IS NOT ON, EXIT TMT.
1059                                    ;       IF THE LOOP FLAG IS ON AND THE LOOP COUNT IS
1060                                    ;       EXHAUSTED, EXIT TMT.
1061                                    ;
1062 005464  032767  000001  172332    3$:     BIT     #LOOP,BASE      ;IS LOOP FLAG ON
1063 005472  001443                            BEQ     7$              ;NO, GET OUT
1064 005474  005767  172464                    TST     LOOPCT          ;IS LOOP COUNT BEING USED
1065 005500  001403                            BEQ     4$              ;NO, JUST KEEP LOOPING
1066 005502  005367  172456                    DEC     LOOPCT          ;SUB FROM LOOP COUNT
1067 005506  001435                            BEQ     7$              ;FINISHED
1068                                    ;
1069                                    ;       IF ALL TESTS, PRINT 'END OF PASS NNNN'.
1070                                    ;       REINITIALIZE FOR NEXT PASS.
1071                                    ;
1072 005510  032767  000010  172306    4$:     BIT     #ALLTST,BASE    ;ALL TESTS
1073 005516  001415                            BEQ     6$
1074 005520  012702  003306                    MOV     #ENDOF,R2       ;POINT TO MESSAGE
1075 005524  012703  000014                    MOV     #ENDLN,R3       ;LENGTH OF MESSAGE
```

```
1076 005530  012705  001465'                    MOV     #PRINT,R5       ;POINT TO PRINT LINE
1077 005534  112225                      5$:    MOVB    (R2)+,(R5)+     ;MOVE MESSAGE TO PRINT LINE
1078 005536  005303                             DEC     R3
1079 005540  001375                             BNE     5$
1080 005542  004767  002742                     JSR     PC,PASSC        ;ADD # PASSES TO PRINT LINE
1081 005546  004767  003130                     JSR     PC,CONSOL       ;PRINT MESSAGE
1082                                     ;
1083 005552  062767  000001  172402      6$:    ADD     #1,PASS         ;COUNT NUMBER OF PASSES
1084 005560  005567  172374                     ADC     PASSH           ;CARRY TO HIGH WORD
1085 005564  012767  000726' 172376             MOV     #MTSUB-2,MTPNT  ;SET POINTER TO CURRENT JUMP TABLE
1086 005572  012767  000111  172372             MOV     #<TESTS*MEM>+1,MTCNT ;SET COUNT
1087 005600  000677                             BR      MTMAIN          ;AND ENTER LOOP
1088                                     ;
1089                                     ;
1090                                     ;       EXIT
1091                                     ;
1092                                     ;
1093 005602  004767  003246              7$:    JSR     PC,ENDTST       ;MEMORY TEST(S) ENDED
1094 005606                              8$:    EXIT$S                  ;AND LEAVE
     005606  012746                             MOV     (PC)+,-(SP)
     005610     063      001                    .BYTE   51.,1
     005612  104377                             EMT     ^O<377>
1095                                     ;
1096                                     ;
1097                                     ;       PREPARE TO JUMP TO MEMORY TEST CONTROL ROUTINES
1098                                     ;
1099                                     ;
1100                                     ;       THE QUOTIENT FROM THE DIVISION BELOW (IN R0) IS THE TEST
1101                                     ;       NUMBER IN ZERO-RELATIVE FORM. THE REMAINDER IS THE MEMORY
1102                                     ;       PLACE-HOLDER VALUE:
1103                                     ;
1104                                     ;       0       FSA A
1105                                     ;       1       FSA B
1106                                     ;       2       FSA C
1107                                     ;       3       INPUT BUFFER
1108                                     ;       4       OUTPUT BUFFER
1109                                     ;       5       BYTE TRANSLATOR
1110                                     ;
1111                                     ;
1112 005614                              JMPMT:
1113 005614  012700  000110                     MOV     #<TESTS*MEM>,R0 ;TOTAL NUMBER OF ROUTINES
1114 005620  166700  172346                     SUB     MTCNT,R0        ;SUB # RTNS REMAINING TO BE EXECUTED
1115 005624  012701  000006                     MOV     #MEM,R1         ;PUT DIVISOR IN R1 FOR SUBRTN
1116 005630  004767  000000G                    JSR     PC,$DIV
1117 005634  010100                             MOV     R1,R0           ;SAVE TABLE POSITION IN WORD OFFSET FORM
1118                                     ;
1119                                     ;       IF TEST 6 IS ABOUT TO BE EXECUTED SET UP THE MEMORY INCREMENT
1120                                     ;       VALUE IN A FIELD THAT THE TEST 6 CONTROL ROUTINE CAN USE.
1121                                     ;
1122 005636  022700  000005                     CMP     #5,R0           ;QUOTIENT = Z-REL TEST NUMBER
1123 005642  001050                             BNE     5$              ;NOT TEST 6 THIS TIME
1124 005644  022701  000005                     CMP     #5,R1           ;BYTE TRANSLATOR
1125 005650  001004                             BNE     1$
1126 005652  012767  000001  172150             MOV     #1,T6FACT       ;SET UP MEMORY INCREMENT
1127 005660  000435                             BR      4$
1128 005662  022701  000004              1$:    CMP     #4,R1           ;OUTPUT BUFFER
1129 005666  001004                             BNE     2$
```

```
005670 012767 000002 172132          MOV   #2,T6FACT            ;SET UP MEMORY INCREMENT
005676 000426                        BR    4$
005700 022701 000003         2$:     CMP   #3,R1
005704 001007                        BNE   3$                   ;INPUT BUFFER
005706 012767 000002 172114          MOV   #2,T6FACT            ;SET UP MEMORY INCREMENT
005714 012767 000200 176360          MOV   #T$DISK,TD$CTW       ;SIMULATE DISK
005722 000445                        BR    LASTJ                ;JUMP TO ROUTINE
005724 012767 000001 172076  3$:     MOV   #1,T6FACT            ;SET UP MEMORY INCREMENT
005732 012767 020000 176360          MOV   #T$SCLK,TD$CTW       ;START CLOCK
005740 010102                        MOV   R1,R2                ;COPY REMAINDER
005742 006302                        ASL   R2                   ;SHIFT FOR WORD OFFSET
005744 016267 000174' 176360         MOV   SETTBL(R2),TD$CTW    ;SELECT WHICH FSA
005752 000431                        BR    LASTJ                ;JUMP TO RTN
005754 012767 000040 176360  4$:     MOV   #T$ICD,TD$CTW        ;SET DIAGNOSTIC BIT
005762 000425                        BR    LASTJ                ;JUMP TO ROUTINE
                             ;
005764 022701 000003         5$:     CMP   #3,R1
005770 001004                        BNE   6$                   ;INPUT BUFFER
005772 012767 000200 176360          MOV   #T$DISK,TD$CTW       ;SIMULATE DISK
006000 000416                        BR    LASTJ
006002 022701 000002         6$:     CMP   #2,R1                ;2 IS MAX OFFSET FOR FSA'S
006006 002410                        BLT   7$                   ;NOT AN FSA
006010 012767 020000 176360          MOV   #T$SCLK,TD$CTW       ;START CLOCK
006016 006301                        ASL   R1                   ;CONVERT TO WORD OFFSET
006020 016167 000174' 176360         MOV   SETTBL(R1),TD$CTW    ;SELECT WHICH FSA
006026 000404                        BR    LASTJ1
006030 012767 000040 176360  7$:     MOV   #T$ICD,TD$CTW        ;SELECT DIAGNOSTIC BIT
                             ;
                             ;       USE THE REMAINDER FROM THE ABOVE DIVISION (X4) AS AN
                             ;       OFFSET INTO THE DOUBLE-WORD MEMORY LIMITS TABLE. MOVE
                             ;       THE 'CURRENT' MEMORY LIMITS TO THE STACK.
                             ;
                             ;       LOAD R0 WITH THE REMAINDER X2 (WORD OFFSET) FOR USE
                             ;       BY THE MEMORY TEST CONTROL ROUTINE.
                             ;
006036 006301                LASTJ:  ASL   R1                   ;SHIFT FOR WORD OFFSET
006040 010100                LASTJ1: MOV   R1,R0                ;LOAD R0
006042 006301                        ASL   R1                   ;SHIFT FOR DBL WORD OFFSET
006044 016146 001150'                MOV   CURLIM(R1),-(SP)     ;MOVE MEMORY END ADDR TO STACK
006050 016146 001152'                MOV   CURLIM+2(R1),-(SP)   ;MOVE MEMORY START ADDRESS
                             ;
006054 016701 172110                 MOV   MTPNT,R1             ;LOAD POINTER TO CONTROL RTN ADDR
006060 000171 000000                 JMP   @(R1)                ;JUMP TO TEST CONTROL ROUTINE
```

```
1174                                    ;
1175                                    ;
1176                                    ;       MEMORY·TEST·CONTROL·ROUTINES·
1177                                    ;
1178                                    ;       TESTS·ARE·DESCRIBED·FULLY·IN·THE·SUB-MODULE·TTEST·
1179                                    ;
1180                                    ;
1181                                    ;       TEST·01
1182·                                   ;
1183                                    ;
1184 006064                            T1:
1185 006064  004770  001214'                   JSR·    PC,@T1ADDR(R0)
1186 006070  000167  000434                    JMP·    MTJUMP·         ;AND·RETURN·TO·LOOP·
1187                                    ;
1188                                    ;
1189                                    ;       TEST·02·
1190                                    ;
1191                                    ;
1192·006074                            T2:
1193 006074  005067  173174·                   CLR·    CKDATA·         ;SET·TEST·PATTERN·TO·ZERO·
1194 006100  004770  001200'                   JSR·    PC,@STADDR(R0)
1195 006104  000167  000420                    JMP·    MTJUMP·
1196                                    ;
1197                                    ;
1198                                    ;       TEST·03
1199                                    ;
1200                                    ;
1201 006110                            T3:
1202·006110  012767  177777  173156          MOV·    #-1,CKDATA·
1203 006116  004770  001200'                   JSR·    PC,@STADDR(R0)
1204 006122·000167  000402·                    JMP·    MTJUMP·
1205                                    ;
1206                                    ;
1207                                    ;       TEST·04
1208                                    ;
1209                                    ;
1210 006126                            T4:
1211 006126  012767  125252  173140          MOV·    #125252,CKDATA
1212·006134  004770  001200'                   JSR·    PC,@STADDR(R0)
1213 006140  000167  000364                    JMP·    MTJUMP·
1214                                    ;
1215                                    ;
1216                                    ;       TEST·05
1217                                    ;
1218                                    ;
1219 006144                            T5:
1220 006144·012767  146314  173122·          MOV··   #146314,CKDATA
1221 006152·004770  001200'                   JSR·    PC,@STADDR(R0)
1222·006156  012767  031463  173110          MOV·    #031463,CKDATA
1223 006164  004770  001200'                   JSR·    PC,@STADDR(R0)
1224 006170  000167  000334                    JMP·    MTJUMP·
1225                                    ;
1226                                    ;
1227                                    ;       TEST·06
1228                                    ;
1229                                    ;
1230 006174                            T6:
```

```
                                        ;       CLR     CKDATA          ;SET PATTERN TO ZERO
                                        ;       JSR     PC,@STADDR(R0)
                                        ;       JSR     PC,@T6ADDR(R0)
                                        ;       ADD     T6FACT,(SP)     ;PERFORM AT NEXT ADDRESS
                                        ;       CLR     CKDATA          ;RESET PATTERN TO ZERO
                                        ;       JSR     PC,@STADDR(R0)
                                        ;       JSR     PC,@T6ADDR(R0)
                                        ;       SUB     T6FACT,(SP)     ;RESTORE ORIGINAL LOWER LIMIT
006174  000167  000330                  ;       JMP     MTJUMP          ;RETURN TO TOP OF MT COMMAND LOOP
                                        ;
                                        ;
                                        ;       TEST 07
                                        ;
                                        ;
006200                                  T7:
006200  004770  001230'                  ;       JSR     PC,@T7ADDR(R0)
006204  000167  000320                  ;       JMP     MTJUMP          ;RETURN TO TOP OF MT COMMAND LOOP
                                        ;
                                        ;
                                        ;       TEST 08
                                        ;
                                        ;
006210                                  T8:
006210  012767  000377  173056          ;       MOV     #377,CKDATA     ;SET TEST PATTERN = X'00FF'
006216  004770  001200'                 ;       JSR     PC,@STADDR(R0)
006222  012767  177400  173044          ;       MOV     #177400,CKDATA  ;SET TEST PATTERN = X'FF00'
006230  004770  001200'                 ;       JSR     PC,@STADDR(R0)
006234  000167  000270                  ;       JMP     MTJUMP          ;RETURN TO TOP OF MT COMMAND LOOP
                                        ;
                                        ;
                                        ;       TEST 09
                                        ;
                                        ;
006240                                  T9:
006240  012767  000001  173026          ;       MOV     #1,CKDATA       ;START TEST PATTERN AT 1
006246  004770  001200'                 1$:     JSR     PC,@STADDR(R0)
006252  016702  173016                  ;       MOV     CKDATA,R2       ;LOAD FOR SHIFT
006256  006302                          ;       ASL     R2              ;SHIFT A BIT
006260  010267  173010                  ;       MOV     R2,CKDATA       ;NEXT TEST PATTERN
006264  005702                          ;       TST     R2              ;FINISHED ?
006266  001367                          ;       BNE     1$
006270  000167  000234                  ;       JMP     MTJUMP          ;RETURN TO LOOP
                                        ;
                                        ;
                                        ;       TEST 10
                                        ;
                                        ;
006274                                  T10:
                                        ;       MOV     MT10,CKDATA     ;USER PATTERN
                                        ;       JSR     PC,@STADDR(R0)
006274  000167  000230                  ;       JMP     MTJUMP          ;RETURN TO TOP OF MT COMMAND LOOP
                                        ;
                                        ;
                                        ;       TEST 11
                                        ;
                                        ;
006300                                  T11:
```

```
1288 006300  012767  100001  172766          MOV     #100001,CKDATA
1289 006306  004770  001200'                 JSR     PC,@STADDR(R0)
1290 006312  012767  040002  172754          MOV     #040002,CKDATA
1291 006320  004770  001200'                 JSR     PC,@STADDR(R0)
1292 006324  012767  020004  172742          MOV     #020004,CKDATA
1293 006332  004770  001200'                 JSR     PC,@STADDR(R0)
1294 006336  012767  001010  172730          MOV     #001010,CKDATA
1295 006344  004770  001200'                 JSR     PC,@STADDR(R0)
1296 006350  012767  004020  172716          MOV     #004020,CKDATA
1297 006356  004770  001200'                 JSR     PC,@STADDR(R0)
1298 006362  012767  002040  172704          MOV     #002040,CKDATA
1299 006370  004770  001200'                 JSR     PC,@STADDR(R0)
1300 006374  012767  001100  172672          MOV     #001100,CKDATA
1301 006402  004770  001200'                 JSR     PC,@STADDR(R0)
1302 006406  012767  000600  172660          MOV     #000600,CKDATA
1303 006414  004770  001200'                 JSR     PC,@STADDR(R0)
1304 006420  000167  000104                 JMP     MTJUMP
1305                                ;
1306                                ;
1307                                ;       TEST 12
1308                                ;
1309                                ;
1310 006424                        T12:
1311 006424  005067  172644                 CLR     CKDATA          ;SET TEST PATTERN
1312 006430  004770  001200'                 JSR     PC,@STADDR(R0)
1313 006434  005067  172636                 CLR     CK2             ;READ PATTERN
1314 006440  012767  177777  172632          MOV     #-1,CK3         ;WRITE PATTERN
1315 006446  004770  001244'                 JSR     PC,@TCDADD(R0)
1316 006452  012767  177777  172616          MOV     #-1,CK2         ;READ PATTERN
1317 006460  005067  172614                 CLR     CK3             ;WRITE PATTERN
1318 006464  004770  001244'                 JSR     PC,@TCDADD(R0)
1319                                ;
1320 006470  005067  172602                 CLR     CK2             ;READ PATTERN
1321 006474  012767  177777  172576          MOV     #-1,CK3         ;WRITE PATTERN
1322 006502  004770  001260'                 JSR     PC,@TCUADD(R0)
1323 006506  012767  177777  172562          MOV     #-1,CK2         ;READ
1324 006514  005067  172560                 CLR     CK3             ;WRITE
1325 006520  004770  001260'                 JSR     PC,@TCUADD(R0)
1326 006524  000167  000000                 JMP     MTJUMP
1327                                ;
1328                                ;
1329 006530  062706  000004        MTJUMP: ADD     #4,SP           ;RESTORE SP
1330 006534  000167  176640                 JMP     MTMAIN          ;AND RETURN
```

```
1332                                    ;
1333                                    ;
1334                                    ;       MEMORY TEST ERROR ROUTINE
1335                                    ;
1336                                    ;
1337 006540                            MEMERR::
1338 006540                                    SAVE    R0,R1,R2,R3,R4,R5
     006540  010046                            MOV     R0,-(SP)
     006542  010146                            MOV     R1,-(SP)
     006544  010246                            MOV     R2,-(SP)
     006546  010346                            MOV     R3,-(SP)
     006550  010446                            MOV     R4,-(SP)
     006552  010546                            MOV     R5,-(SP)
1339                                    ;
1340                                    ;       SET FLAG FOR ERROR ENCOUNTERED, AN ERROR MESSAGE LIMIT
1341                                    ;       COUNT OF ZERO MEANS THAT THE COUNT IS NOT BEING USED,
1342                                    ;       A COUNT OF -1 MEANS THAT THE LIMIT HAS BEEN REACHED
1343                                    ;       (NO MORE ERROR MESSAGES ARE TO BE PRINTED).
1344                                    ;
1345 006554  052767  000040  171242            BIS     #ERROR,BASE     ;SET FLAG FOR ERROR ENCOUNTERED
1346 006562  005767  171226                    TST     ERLIM
1347 006566  001412                            BEQ     2$
1348 006570  003002                            BGT     1$
1349 006572  000167  000372                    JMP     ERRORX
1350                                    ;
1351                                    ;       DECREMENT ERROR LIMIT COUNT, IF IT GOES ZERO HERE, SET
1352                                    ;       NO MORE ERROR MESSAGES.
1353                                    ;       IT TO -1.
1354                                    ;
1355 006576  005367  171212            1$:     DEC     ERLIM
1356 006602  001004                            BNE     2$
1357 006604  005367  171204                    DEC     ERLIM
1358 006610  000167  000354                    JMP     ERRORX
1359                                    ;
1360                                    ;
1361                                    ;       DERIVE TEST NUMBER FROM THE ADDRESS POSITION OF THE CURRENT
1362                                    ;       MEMORY TEST CONTROL ROUTINE IN THE CURRENT JUMP TABLE.
1363                                    ;
1364                                    ;       GET THE OFFSET FROM THE BEGINNING OF THE TABLE, DIVIDE
1365                                    ;       BY THE NUMBER OF MEMORIES IN THE TABLE, THE QUOTIENT
1366                                    ;       IS THE TEST NUMBER (ZERO-RELATIVE), THE REMAINDER IS THE
1367                                    ;       MEMORY PLACE-HOLDER VALUE, SAVE IT.
1368                                    ;
1369 006614  012700  000110            2$:     MOV     #<TESTS*MEM>,R0 ;LOAD NUMBER OF TESTS (TOTAL)
1370 006620  166700  171346                    SUB     MTCNT,R0        ;SUB CURRENT COUNT OF TABLE ENTRIES
1371 006624  012701  000006                    MOV     #MEM,R1         ;SYSTEM SUBRTN EXPECTS DIVISOR IN R1
1372 006630  004767  000000G                   JSR     PC,$DIV         ;DIVIDEND IS IN R1
1373 006634  010146                            MOV     R1,-(SP)        ;TEMP SAVE REMAINDER
1374 006636  010001                            MOV     R0,R1           ;PREPARE FOR CONVERSION TO DECIMAL
1375 006640  005201                            INC     R1              ;ADJUST ZERO RELATIVE NUMBER
1376 006642  012700  001316'                   MOV     #ASWRK,R0       ;CONVERT INTO WORK FIELD
1377 006646  012702  000001                    MOV     #1,R2           ;SUPPLY LEADING ZEROS
1378 006652  004767  0000000G                  JSR     PC,$CBDSG       ;CONVERT BINARY TO DECIMAL
1379                                    ;
1380 006656  012705  001465'                   MOV     #PRINT,R5       ;POINT AT PRINT LINE
1381 006662  012700  001323'                   MOV     #TMSG,R0        ;POINT AT 'TEST '
1382 006666  012701  000005                    MOV     #5,R1           ;NUMBER OF BYTES IN STRING
```

```
1383 006672  112025                3$:     MOVB    (R0)+,(R5)+     ;MOVE LABEL TO PRINT LINE
1384 006674  005301                         DEC     R1
1385 006676  001375                         BNE     3$
1386                                ;
1387 006700  116725  172415                 MOVB    ASWRK+3,(R5)+   ;MOVE A DIGIT OF TEST NUMBER
1388 006704  116725  172412                 MOVB    ASWRK+4,(R5)+
1389 006710  062705  000002                 ADD     #2,R5           ;BUMP PRINT LINE POINTER
1390                                ;
1391                                ;       MOVE PASS COUNT TO PRINT LINE.
1392                                ;
1393 006714  012700  001330'                MOV     #PMSG,R0        ;POINT AT 'PASS '
1394 006720  012701  000005                 MOV     #5,R1           ;NUMBER OF BYTES IN STRING
1395 006724  112025                4$:     MOVB    (R0)+,(R5)+     ;MOVE LABEL TO PRINT LINE
1396 006726  005301                         DEC     R1
1397 006730  001375                         BNE     4$
1398 006732  004767  001552                 JSR     PC,PASSC        ;CONVERT NUMBER OF PASSES
1399 006736  062705  000003                 ADD     #3,R5           ;BUMP POINTER
1400                                ;
1401                                ;       REPORT 'FAILING MEMORY'
1402                                ;
1403 006742  012700  001335'                MOV     #FMSG,R0        ;POINT AT STRING
1404 006746  012701  000020                 MOV     #16.,R1         ;LENGTH OF STRING
1405 006752  112025                5$:     MOVB    (R0)+,(R5)+     ;MOVE LABEL TO PRINT LINE
1406 006754  005301                         DEC     R1
1407 006756  001375                         BNE     5$
1408                                ;
1409                                ;       USE THE REMAINDER FROM THE ABOVE DIVISION AS AN INDEX
1410                                ;       INTO A TABLE OF MEMORY NAMES (=6 BYTES EACH).
1411                                ;
1412 006760  012601                         MOV     (SP)+,R1        ;LOAD REMAINDER FROM DIVISION ABOVE
1413 006762  012700  000006                 MOV     #6,R0           ;SYSTEM SUBRTN EXPECTS MULTIPLIER IN R0
1414 006766  004767  000000G                JSR     PC,$MUL
1415 006772  012700  001417'                MOV     #FTBL,R0        ;POINT TO TABLE OF MEMORY NAMES
1416 006776  060100                         ADD     R1,R0           ;POINT TO THE ONE THAT FAILED
1417 007000  012701  000006                 MOV     #6,R1           ;NUMBER OF CHARS IN NAME
1418 007004  112025                6$:     MOVB    (R0)+,(R5)+     ;MOVE NAME TO PRINT LINE
1419 007006  005301                         DEC     R1
1420 007010  001375                         BNE     6$
1421 007012  004767  001664                 JSR     PC,CONSOL       ;WRITE ONE LINE TO CONSOLE
1422                                ;
1423                                ;       LINE 2
1424                                ;       ADDRESS OF ERROR, EXPECTED PATTERN, RECEIVED PATTERN
1425                                ;
1426 007016  012705  001465'                MOV     #PRINT,R5       ;POINT TO PRINT LINE
1427 007022  012700  001355'                MOV     #AMSG,R0        ;POINT TO 'ADDRESS '
1428 007026  012701  000011                 MOV     #9.,R1          ;LOAD LENGTH OF STRING
1429 007032  112025                7$:     MOVB    (R0)+,(R5)+     ;MOVE LABEL
1430 007034  005301                         DEC     R1
1431 007036  001375                         BNE     7$
1432                                ;
1433 007040  016701  172240                 MOV     ERRADD,R1       ;LOAD ERROR ADDRESS
1434 007044  004767  001404                 JSR     PC,UNPK         ;CONVERT TO PBINTABLE CHARS
1435 007050  005205                         INC     R5              ;BUMP PRINT LINE POINTER
1436                                ;
1437 007052  012700  001366'                MOV     #EMSG,R0        ;POINT TO 'EXPECTED'
1438 007056  012701  000012                 MOV     #10.,R1         ;LOAD NUMBER OF CHARS
1439 007062  112025                8$:     MOVB    (R0)+,(R5)+     ;MOVE LABEL
```

```
     007064  005301                     DEC     R1
     007066  001375                     BNE     8$
                                ;
     007070  016701  172200             MOV     CKDATA,R1          ;LOAD MEMMORY TEST PATTERN
     007074  004767  001354             JSR     PC,UNPK            ;CONVERT
     007100  005205                     INC     R5                 ;BUMP PRINT LINE POINTER
                                ;
     007102  012700  001400'            MOV     #RMSG,R0           ;POINT TO 'RECEIVED '
     007106  012701  000012             MOV     #10.,R1            ;LOAD NUMBER OF CHARS
     007112  112025             9$:     MOVB    (R0)+,(R5)+        ;MOVE LABEL
     007114  005301                     DEC     R1
     007116  001375                     BNE     9$
                                ;
                                ;       COUNT OF NUMBER OF WORDS TO PRINT = 0 SIGNALS THAT
                                ;       THE RESULTS OF SEVERAL MEMORY READS WERE UNPREDICTABLE;
                                ;       PRINT * * *
                                ;
     007120  016700  172162             MOV     ERRCT,R0           ;LOAD NUMBER OF WORDS TO CONVERT
     007124  003010                     BGT     11$                ;PRINT WORDS
     007126  012700  001412'            MOV     #UNMSG,R0          ;* * *
     007132  012701  000005             MOV     #5,R1              ;LENGTH OF MESSAGE
     007136  112025             10$:    MOVB    (R0)+,(R5)+
     007140  005301                     DEC     R1
     007142  001375                     BNE     10$
     007144  000407                     BR      13$
                                ;
     007146  012702  001310'    11$:    MOV     #ERW1,R2           ;POINT TO FIRST OF THEM
     007152  012201             12$:    MOV     (R2)+,R1           ;LOAD THE WORD ITSELF
     007154  004767  001274             JSR     PC,UNPK
     007160  005300                     DEC     R0
     007162  001373                     BNE     12$
                                ;
     007164  004767  001512     13$:    JSR     PC,CONSOL          ;ELSE WRITE TO CONSOLE
                                ;
     007170                     ERRORX:
     007170                             RESTOR  R0,R1,R2,R3,R4,R5
     007170  012605                     MOV     (SP)+,R5
     007172  012604                     MOV     (SP)+,R4
     007174  012603                     MOV     (SP)+,R3
     007176  012602                     MOV     (SP)+,R2
     007200  012601                     MOV     (SP)+,R1
     007202  012600                     MOV     (SP)+,R0
     007204  000207                     RTS     PC
```

```
                                        ;
                                        ;
                                        ;       AST·
                                        ;
                                        ;       INTERCEPT·CHARACTER·FROM·CONSOLE.
                                        ;       ON 'W' PRINT TEST·NUMBER, PASS·COUNT, MEMORY·NAME, ADDRESS,
                                        ;       AND·TEST·PATTERN.
                                        ;       ON 'C' PRINT TEST·PATTERN·('CKDATA').
                                        ;       ON 'P' PRINT NUMBER·OF PASSES.
                                        ;       ON 'T' PRINT TEST·NUMBER.
                                        ;       ON ALL·OTHERS·EXIT·TMT.
                                        ;
                                        ;
007206                                  AST:
007206  012667  170572                          MOV·    (SP)+,ASTWRD·   ;GET·CHAR·
007212·                                         SAVE·   R0,R1,R2,R3,R4,R5
007212·010046                                   MOV·    R0,-(SP)
007214 010146                                   MOV·    R1,-(SP)
007216 010246                                   MOV·    R2,-(SP)
007220 010346                                   MOV·    R3,-(SP)
007222·010446                                   MOV·    R4,-(SP)
007224 010546                                   MOV·    R5,-(SP)
                                        ;
007226  122767  000120  170550                  CMPB·   #'P,ASTWRD·     ;PRINT·NUMBER·OF·PASSES·AND·CONTINUE·
007234  001002·                                 BNE·    1$
007236 000167  000336                           JMP·    PAST
007242·122767  000103  170534  1$:             CMPB·   #'C,ASTWRD·     ;PRINT·OUT·TEST·PATTERN·
007250 001002·                                  BNE·    2$
007252·000167  000202                           JMP·    CAST
007256 122767  000124  170520  2$:             CMPB·   #'T,ASTWRD·     ;PRINT·OUT·TEST·NUMBER·
007264·001002·                                  BNE·    3$
007266 000167  000210                           JMP·    TAST
007272·122767  000127  170504  3$:             CMPB·   #'W,ASTWRD·     ;PRINT·WHERE·
007300 001402·                                  BEQ·    WAST
007302·000167  000256                           JMP·    EAST
                                        ;
                                        ;
                                        ;       ROUTINE·FOR·'W'
                                        ;
                                        ;       DERIVE·TEST·NUMBER·FROM·THE·POSITION·OF·THE·CURRENT·
                                        ;       MEMORY·TEST·CONTROL·ROUTINE·IN·THE·CURRENT JUMP·TABLE.
                                        ;
                                        ;       GET·THE·OFFSET·FROM·THE·BEGINNING·OF·THE·TABLE. DIVIDE·
                                        ;       BY·THE·NUMBER·OF·MEMORIES·IN·THE TABLE. THE·QUOTIENT
                                        ;       IS·THE·TEST·NUMBER·(ZERO-RELATIVE). THE·REMAINDER·IS THE·
                                        ;       MEMORY·PLACE-HOLDER·VALUE. SAVE·IT.
                                        ;
                                        ;
007306                                  WAST:
007306  012700  000110                          MOV·    #<TESTS*MEM>,R0 ;LOAD·NUMBER·OF·TESTS·(TOTAL)
007312·166700  170654                           SUB·    MTCNT,R0        ;SUB·CURRENT·ROUTINE·NUMBER·
007316 012701  000006                           MOV·    #MEM,R1         ;SYSTEM·SUBRTN·EXPECTS·DIVISOR·IN·R1
007322·004767  000000G                          JSR·    PC,$DIV·        ;DIVIDEND·IS·IN·R1
007326 010146                                   MOV·    R1,-(SP)        ;TEMP SAVE REMAINDER
007330 010001                                   MOV·    R0,R1           ;PREPARE FOR CONVERSION TO DECIMAL
007332·005201                                   INC·    R1              ;ADJUST·ZERO·RELATIVE·NUMBER
007334 012700  001316'                          MOV·    #ASWRK,R0       ;CONVERT·INTO·WORK·FIELD·
```

```
007340  012702  000001          MOV     #1,R2           ;SUPPLY LEADING ZEROS
007344  004767  000000G         JSR     PC,$CBDSG       ;CONVERT BINARY TO DECIMAL
                        ;
                        ;       TEST NUMBER
                        ;
007350  012705  001465'         MOV     #PRINT,R5       ;R5 -> PRINT LINE
007354  116725  171741          MOVB    ASWRK+3,(R5)+   ;MOVE 1 DIGIT OF TEST NUMBER
007360  116725  171736          MOVB    ASWRK+4,(R5)+
007364  005205                  INC     R5
                        ;
                        ;       PASS
                        ;
007366  004767  001116          JSR     PC,PASSC        ;PRINT OUT PASS NUMBER
007372  005205                  INC     R5              ;BUMP PRINT LINE POINTER
                        ;
                        ;       MEMORY
                        ;
                        ;       USE THE REMAINDER FROM THE ABOVE DIVISION AS AN INDEX
                        ;       INTO A TABLE OF MEMORY NAMES.
                        ;
007374  012601          3$:     MOV     (SP)+,R1        ;LOAD REMAINDER FROM DIVISION ABOVE
007376  012700  000006          MOV     #6,R0           ;SYSTEM SUBRTN EXPECTS MULTIPLIER IN R0
007402  004767  000000G         JSR     PC,$MUL
007406  012700  001417'         MOV     #FTBL,R0        ;POINT TO TABLE OF MEMORY NAMES
007412  060100                  ADD     R1,R0           ;POINT TO THE ONE THAT FAHLED
007414  012701  000006          MOV     #6,R1           ;NUMBER OF CHARS IN NAME
007420  112025          4$:     MOVB    (R0)+,(R5)+     ;MOVE NAME TO PRINT LINE
007422  005301                  DEC     R1
007424  001375                  BNE     4$
007426  005205                  INC     R5
                        ;
                        ;       ADDRESS
                        ;
007430  016701  171646          MOV     PREADD,R1       ;LOAD ERROR ADDRESS
007434  004767  001014          JSR     PC,UNPK         ;CONVERT TO PBINTABLE CHARS
007440  005205                  INC     R5              ;BUMP PRINT LINE POINTER
                        ;
                        ;       TEST PATTERN
                        ;
007442  016701  171626          MOV     CKDATA,R1       ;LOAD MEMMORY TEST PATTERN
007446  004767  001002          JSR     PC,UNPK         ;CONVERT
007452  004767  001224          JSR     PC,CONSOL       ;ELSE WRITE TO CONSOLE
007456  000475                  BR      REST            ;EXIT AST
                        ;
                        ;
                        ;       PRINT TEST PATTERN ALONE
                        ;
                        ;
007460                  CAST:
007460  012705  001465'         MOV     #PRINT,R5
007464  016701  171604          MOV     CKDATA,R1       ;LOAD TEST PATTERN
007470  004767  000760          JSR     PC,UNPK         ;CONVERT TO ASCII
007474  004767  001202          JSR     PC,CONSOL       ;WRITE TO TT0
007500  000464                  BR      REST
                        ;
                        ;
                        ;       PRINT OUT TEST NUMBER ALONE
```

```
                                        ;
                                        ;
                                        ;       DERIVE TEST NUMBER FROM THE POSITION OF THE CURRENT
                                        ;       MEMORY TEST CONTROL ROUTINE IN THE CURRENT JUMP TABLE.
                                        ;
                                        ;       GET THE OFFSET FROM THE BEGINNING OF THE TABLE, DIVIDE
                                        ;       BY THE NUMBER OF MEMORIES IN THE TABLE; THE QUOTIENT
                                        ;       IS THE TEST NUMBER (ZERO-RELATIVE).
                                        ;
                                        ;
007502                                  TAST:
007502  012700  000110'                        MOV     #<TESTS*MEM>,R0 ;LOAD NUMBER OF TESTS (TOTAL)
007506  166700  170460'                       SUB     MTCNT,R0        ;SUB CURRENT ROUTINE NUMBER
007512  012701  000006                        MOV     #MEM,R1         ;SYSTEM SUBRTN EXPECTS DIVISOR IN R1
007516  004767  000000G                       JSR     PC,$DIV         ;DIVIDEND IS IN R1
007522  010001                                MOV     R0,R1           ;PREPARE FOR CONVERSION TO DECIMAL
007524  005201                                INC     R1              ;ADJUST ZERO RELATIVE NUMBER
007526  012700  001316'                       MOV     #ASWRK,R0       ;CONVERT INTO WORK FIELD
007532  012702  000001                        MOV     #1,R2           ;SUPPLY LEADING ZEROS
007536  004767  000000G                       JSR     PC,$CBDSG       ;CONVERT BINARY TO DECIMAL
                                        ;
007542  012705  001465'                       MOV     #PRINT,R5       ;R5 -> PRINT LINE
007546  116725  171547                        MOVB    ASWRK+3,(R5)+   ;MOVE 1 DIGIT OF TEST NUMBER
007552  116725  171544                        MOVB    ASWRK+4,(R5)+
007556  004767  001120                        JSR     PC,CONSOL
007562  000433                                BR      REST
                                        ;
                                        ;
                                        ;
                                        ;       EXIT PROGRAM
                                        ;
                                        ;
                                        ;       IF THE 'ALL TESTS, ALL MEMORIES, FULL RANGE' OPTION
                                        ;       IS IN EFFECT, THE PASS NUMBER FOR THE PREVIOUS PASS
                                        ;       HAS ALREADY BEEN PRINTED OUT (SEE 'MTMAIN').
                                        ;
007564                                  EAST:
007564  004767  001270                        JSR     PC,HLTTST       ;WRITE HALT MESSAGE
007570  032767  000010  170226                BIT     #ALLTST,BASE    ;ALL TESTS ?
007576  001025                                BNE     REST
                                        ;
                                        ;
                                        ;       PRINT NUMBER OF PASSES ALONE
                                        ;
                                        ;
007600                                  PAST:
007600  012700  001465'                       MOV     #PRINT,R0       ;POINT TO PRINT LINE
007604  012701  000116                        MOV     #78.,R1         ;NUMBER OF CHAR POSITIONS
007610  112720  000040                1$:     MOVB    #40,(R0)+       ;CLEAR THE LINE
007614  005301                                DEC     R1
007616  001374                                BNE     1$
                                        ;
007620  012705  001465'                       MOV     #PRINT,R5       ;POINT TO PRINT LINE
007624  012700  003272'                       MOV     #PMSG2,R0       ;POINT TO 'NUMBER OF PASSES'
007630  012701  000014                        MOV     #PM2LN,R1       ;LENGTH OF STRING
007634  112025                        2$:     MOVB    (R0)+,(R5)+     ;MOVE STRING TO PRINT LINE
007636  005301                                DEC     R1
```

```
1643 007640  001375                          BNE·····  2$
1644                                 ;
1645 007642· 004767  000642·                 JSR·      PC,PASSC·          ;CONVERT DOUBLE·WORD
1646 007646  004767  001030·                 JSR·      PC,CONSOL·
1647                                 ;
1648 007652                          REST:   RESTOR·   R0,R1,R2,R3,R4,R5
     007652· 012605                          MOV·      (SP)+,R5
     007654  012604                          MOV·      (SP)+,R4
     007656  012603                          MOV·      (SP)+,R3
     007660  012602·                         MOV·      (SP)+,R2·
     007662· 012601                          MOV·      (SP)+,R1
     007664  012600                          MOV·      (SP)+,R0
1649                                 ;
1650 007666  122767  000120·  170110        CMPB·     #'P,ASTWRD·        ;PRINT·# PASSES·AND·CONTINUE
1651 007674  001414                          BEQ·      10$
1652·007676  122767  000103·  170100        CMPB·     #'C,ASTWRD·        ;PRINT·TEST·PATTERN·AND·CONTINUE·
1653 007704  001410                          BEQ·      10$
1654 007706  122767  000124   170070        CMPB·     #'T,ASTWRD·        ;PRINT·TEST·NUMBER AND·CONTINUE·
1655 007714  001404                          BEQ·      10$
1656 007716  122767  000127·  170060        CMPB·     #'W,ASTWRD·        ;PRINT·WHERE·AND·CONTINUE·
1657 007724  001003                          BNE·      1$                 ;NO, EXIT·
1658                                 ;
1659 007726                          10$:    ASTX$S·
     007726  012746                          MOV·      (PC)+,-(SP)
     007730     163      001·                .BYTE·    115.,1
     007732· 104377                          EMT·      ^O<377>
1660                                 ;
1661 007734  012701  001465'                1$:       MOV·      #PRINT,R1          ;POINT·TO·PRINT·LINE·
1662·007740  012700  000116                  MOV·      #78.,R0            ;NUMBER·OF·BYTES·
1663 007744  112711  000040·                2$:       MOVB      #40.,(R1)          ;CLEAR·LINE·TO·BLANKS
1664 007750  005300                          DEC·      R0
1665 007752· 001374                          BNE·      2$
1666 007754  112767  000015·  171503        MOVB·     #15,PRINT·         ;WRITE·OUT·1 CR+LF
1667 007762· 112767  000012·  171476        MOVB·     #12,PRINT+1
1668 007770  004767  000706·                 JSR·      PC,CONSOL·
1669                                 ;
1670 007774                          EXIT$S·
     007774  012746                          MOV·      (PC)+,-(SP)
     007776     063      001                .BYTE·    51.,1
     010000  104377                          EMT·      ^O<377>
```

```
1672                                    ;
1673                                    ;
1674                                    ;       SUBRTN FOR MEMORY LIMITS
1675                                    ;
1676                                    ;       INPUT:
1677                                    ;       2(SP)   ABSOLUTE LOWER LIMITS
1678                                    ;       4(SP)   ABSOLUTE UPPER LIMITS
1679                                    ;       6(SP)   ADDRESS OF PROMPT ROUTINE
1680                                    ;
1681                                    ;       OUTPUT:
1682                                    ;       (SP)    CURRENT WORKING LOWER LIMITS
1683                                    ;       2(SP)   CURRENT WORKING UPPER LIMITS
1684                                    ;
1685                                    ;       WORK FIELDS USED:
1686                                    ;       LOWER
1687                                    ;       UPPER
1688                                    ;
1689                                    ;       PROMPT FOR LIMITS. A <CR> RESPONSE MEANS TO TAKE
1690                                    ;       THE ABSOLUTE LOWER AND UPPER LIMITS AND RETURN
1691                                    ;       THEM ON THE STACK. OTHERWISE IF THE RESPONSE IS IN
1692                                    ;       THE FORM:
1693                                    ;
1694                                    ;       >000000 000012
1695                                    ;
1696                                    ;       THIS ROUTINE CONVERTS THE FIRST VALUE AND COMPARES IT
1697                                    ;       AGAINST THE MEMORY'S ABSOLUTE LOWER LIMITS AT 2(SP). IF
1698                                    ;       THE NEW LIMITS ARE IN RANGE, THEY ARE PLACED IN A
1699                                    ;       TEMPORARY WORK FIELD. THE ROUTINE THEN CHECKS THE COMMAND
1700                                    ;       LINE FOR THE UPPER LIMITS, CONVERTS THEM, AND COMPARES
1701                                    ;       THEM AGAINST THE MEMORY'S ABSOLUTE UPPER LIMITS. IF THE
1702                                    ;       NEW LIMITS ARE IN RANGE, THEY ARE PLACE IN A TEMPORARY
1703                                    ;       WORK AREA. THE ROUTINE THEN COMPARES THE NEW UPPER LIMITS
1704                                    ;       WITH THE NEW LOWER LIMITS. IF THE NEW UPPER LIMITS ARE
1705                                    ;       EQUAL TO OR GREATER THAN THE NEW LOWER LIMITS, BOTH NEW
1706                                    ;       VALUES ARE PLACED ON THE STACK; THE ROUTINE THAT CALLED
1707                                    ;       LIMITS WILL TAKE THESE VALUES OFF THE STACK AND PLACE
1708                                    ;       THEM IN THE 'CURRENT LIMITS TABLE'. DURING THE MEMORY
1709                                    ;       TEST CYCLE, THE LIMITS FROM THIS TABLE ARE MADE AVAILABLE
1710                                    ;       TO THE MEMORY TEST ROUTINES.
1711                                    ;
1712                                    ;
1713 010002                            LIMITS:
1714 010002  004776  000006                    JSR     PC,@6(SP)       ;PROMPT FOR LIMITS
1715 010006  004767  000262                    JSR     PC,FIND         ;FIRST FIND A NUMBER
1716 010012  103472                            BCS     LIMX2           ;NO OVERRIDES, LEAVE LIMITS ALONE
1717 010014  004767  000372                    JSR     PC,PACK         ;CONVERT LOWER LIMIT TO BINARY
1718 010020  103003                            BCC     1$              ;VALUE OK, CONTINUE
1719 010022  004767  001112                    JSR     PC,ERR4
1720 010026  000765                            BR      LIMITS          ;TRY AGAIN
1721                                    ;
1722                                    ;       CHECK LOWER LIMITS
1723                                    ;
1724 010030  026766  167762  000002  1$:       CMP     BINWD,2(SP)     ;COMPARE LOWER LIMITS
1725 010036  103003                            BHIS    2$              ;OK, CONTINUE
1726 010040  004767  001074                    JSR     PC,ERR4
1727 010044  000756                            BR      LIMITS          ;TRY AGAIN
1728                                    ;
```

```
                                        ;       ALSO CHECK NEW LOWER LIMITS AGAINST ABSOLUTE UPPER LIMITS
                                        ;       ON THE STACK. AN ERROR HERE WOULD SHOW UP BELOW BUT IT
                                        ;       IS MORE CORRECT TO REPORT AN ERROR IN LOWER LIMITS IF
                                        ;       IF THE NEW LOWER LIMITS ARE HIGHER THAN THE ABSOLUTE
                                        ;       UPPER LIMITS.
                                        ;
010046  026766  167744  000004  2$:     CMP     BINWD,4(SP)     ;IS NEW LOW LIMIT GT ABS UPPER LIMIT
010054  101403                          BLOS    20$             ;NO, NEW LOWER LIMIT IS OK
010056  004767  001056                  JSR     PC,ERR4
010062  000747                          BR      LIMITS
                                        ;
                                        ;       SAVE NEW LOWER LIMITS, FIND NEW UPPER LIMITS IN COMMAND
                                        ;       LINE.
                                        ;
010064  016767  167726  167730  20$:    MOV     BINWD,LOWER     ;MOVE IN NEW LOW LIMITS
010072  004767  000176                  JSR     PC,FIND         ;FIND UPPER LIMITS IN COMMAND LINE
010076  103003                          BCC     3$              ;OK, CONTINUE
010100  004767  001030                  JSR     PC,ERR5
010104  000736                          BR      LIMITS          ;START OVER
                                        ;
010106  004767  000300          3$:     JSR     PC,PACK         ;CONVERT UPPER LIMITS
010112  103003                          BCC     4$              ;OK, CONTINUE
010114  004767  001014                  JSR     PC,ERR5
010120  000730                          BR      LIMITS          ;START OVER
                                        ;
                                        ;       CHECK NEW UPPER LIMITS.
                                        ;
010122  026766  167670  000004  4$:     CMP     BINWD,4(SP)     ;COMPARE UPPER LIMITS
010130  101403                          BLOS    5$              ;OK, CONTINUE
010132  004767  000776                  JSR     PC,ERR5
010136  000721                          BR      LIMITS          ;START OVER
                                        ;
                                        ;       COMPARE NEW LOWER LIMITS WITH NEW UPPER LIMITS.
                                        ;
010140  016767  167652  167652  5$:     MOV     BINWD,UPPER     ;PLACE NEW UPPER LIMIT ON STACK
010146  026767  167650  167644          CMP     LOWER,UPPER     ;IS UPPER LIMIT GT LOWER LIMIT
010154  101403                          BLOS    LIMX            ;YES, EXIT
010156  004767  000752                  JSR     PC,ERR5
010162  000707                          BR      LIMITS          ;TRY AGAIN
                                        ;
010164  016766  167632  000002  LIMX:   MOV     LOWER,2(SP)     ;PUT NEW LOWER LIMITS ON STACK
010172  016766  167622  000004          MOV     UPPER,4(SP)     ;PUT NEW UPPER LIMITS ON STACK
010200  000207                  LIMX2:  RTS     PC
```

```
                                        ;
                                        ;
                                        ;       SCAN A TABLE FOR A VALID COMMAND/MNEMONIC
                                        ;
                                        ;       INPUT:
                                        ;       R0 = NUMBER OF ENTRIES IN COMMAND TABLE
                                        ;       R1 -> CHAR STRING IN GCML COMMAND LINE
                                        ;       R2 -> TOP OF COMMAND TABLE
                                        ;
                                        ;       OUTPUT:
                                        ;       R1 -> ROUTINE THAT GOVERNS THE COMMAND (IF MATCH WAS MADE)
                                        ;       R1 -> CHAR STRING IN COMMAND LINE (IF NO MATCH WAS MADE)
                                        ;       R0 = RELATIVE POSITION OF MATCHED ENTRY IN TABLE
                                        ;
                                        ;
1788 010202                             SCAN:
1789 010202  010346                             MOV     R3,-(SP)        ;SAVE R3
1790 010204  010046                             MOV     R0,-(SP)        ;SAVE # ENTRIES
1791 010206  010146                             MOV     R1,-(SP)        ;SAVE POINTER TO BEGINNING OF STRING
1792                                    ;
1793 010210  011601                     FNOUT1: MOV     (SP),R1         ;POINT TO NON-BLANK IN COMMAND LINE
1794 010212  012703  000002                     MOV     #2,R3           ;NUMBER OF CHARS IN NON-BLANK FIELD
1795 010216  122122                     FNIN1:  CMPB    (R1)+,(R2)+     ;DOES COMMAND LINE MATCH TABLE ENTRY
1796 010220  001003                             BNE     FNOUT2          ;NO, TRY NEXT TABLE ENTRY
1797 010222  005303                             DEC     R3              ;SUB FROM LOOP COUNT
1798 010224  001374                             BNE     FNIN1
1799 010226  000411                             BR      FNMTCH          ;COMMAND FOUND IN TABLE
1800 010230  060302                     FNOUT2: ADD     R3,R2           ;ADD # UNCOMPARED CHARS TO POINTER
1801 010232  005202                             INC     R2              ;THEN ADJUST TO NEXT TABLE ENTRY
1802 010234  005300                             DEC     R0              ;SUB FROM OUTER LOOP COUNT
1803 010236  001364                             BNE     FNOUT1          ;TRY AGAIN
1804 010240  012601                             MOV     (SP)+,R1        ;RESTORE COMMAND LINE POINTER
1805 010242  012600                             MOV     (SP)+,R0        ;RESTORE R0
1806 010244  012603                             MOV     (SP)+,R3        ;RESTORE R3
1807 010246  000261                             SEC                     ;COMMAND NOT IN TABLE
1808 010250  000207                             RTS     PC
1809                                    ;
1810 010252  010201                     FNMTCH: MOV     R2,R1           ;POINT R1 AT MEMORY FLAG IN TABLE
1811 010254  062706  000002                     ADD     #2,SP           ;POINT TO INCOMING R0 ON STACK
1812 010260  012602                             MOV     (SP)+,R2        ;GET TOTAL # TABLE ENTRIES
1813 010262  160002                             SUB     R0,R2           ;GET POSITION OF MATCHED ENTRY
1814 010264  010200                             MOV     R2,R0           ;PUT IN R0 FOR RETURN
1815 010266  012603                             MOV     (SP)+,R3        ;RESTORE R3
1816 010270  000241                             CLC
1817 010272  000207                             RTS     PC
```

```
                                ;
                                ;
                                ;       FIND THE NEXT NON-BLANK IN THE COMMAND BUFFER,
                                ;       THEN FIND THE LENGTH OF THE STRING THAT STARTS WITH THAT CHARACTER.
                                ;
                                ;       INPUT:
                                ;       GCMLEN  -  NUMBER OF UNPROCESSED BYTES IN COMMAND LINE
                                ;       GCMPNT  - ADDR OF NEXT UNPROCESSED POSITION IN COMMAND LINE
                                ;
                                ;       OUTPUT:
                                ;       R1 -> STRING, R0 = LENGTH OF STRING
                                ;       GCMLEN, GCMPNT UPDATED FOR NEXT ENTRY INTO THIS ROUTINE
                                ;
                                ;       THIS ROUTINE IS DESIGNED TO BE ENTERED A NUMBER OF TIMES
                                ;       IN THE PARSING OF A COMMAND LINE.  THE FIELDS GCMLEN AND
                                ;       GCMPNT ARE REFRESHED WHEN A NEW COMMAND LINE IS READ
                                ;       (SEE THE MESSAGE PRINTING/PROMPTING CODE).
                                ;
                                ;
010274                          FIND:
010274  010246                          MOV     R2,-(SP)        ;SAVE R2
010276  016701  167652                  MOV     GCMLEN,R1       ;# BYTES REMAINING IN COMMAND BUFFER
010302  001440                          BEQ     FSECX           ;THERE ARE NONE
010304  016702  167646                  MOV     GCMPNT,R2       ;LOAD CURRENT POINTER
010310  122712  000040          1$:     CMPB    #40,(R2)        ;LOOK FOR A BLANK
010314  001403                          BEQ     10$             ;OK, BUMP TO NEXT CHAR
010316  122712  000054                  CMPB    #',,(R2)        ;COMMA IN COMMAND LINE
010322  001004                          BNE     2$              ;TREAT COMMA AS BLANK
010324  005202                  10$:    INC     R2              ;BUMP POINTER
010326  005301                          DEC     R1              ;SUB FROM REMAINING LENGTH
010330  001367                          BNE     1$
010332  000424                          BR      FSECX           ;NO NON-BLANK FOUND
                                ;
010334  010246                  2$:     MOV     R2,-(SP)        ;TEMP SAFE POINTER TO BEGINNING OF STRING
010336  005000                          CLR     R0              ;CLEAR CHAR COUNT
010340  122712  000040          3$:     CMPB    #40,(R2)        ;LOOK FOR A BLANK
010344  001407                          BEQ     4$              ;FOUND END OF STRING
010346  122712  000054                  CMPB    #',,(R2)        ;TREAT COMMAS AS BLANKS
010352  001404                          BEQ     4$
010354  005202                          INC     R2              ;BUMP POINTER
010356  005200                          INC     R0              ;BUMP CHAR COUNT
010360  005301                          DEC     R1              ;SUB FROM BYTES REMAINING
010362  001366                          BNE     3$
                                ;
010364  010267  167566          4$:     MOV     R2,GCMPNT       ;SAVE POINTER FOR NEXT TIME
010370  010167  167560                  MOV     R1,GCMLEN       ;SAVE BYTES REMAINING FOR NEXT TIME
010374  012601                          MOV     (SP)+,R1        ;POINTER TO BEGINNING OF STRING
010376  012602                          MOV     (SP)+,R2        ;RESTORE R2
010400  000241                          CLC
010402  000207                          RTS     PC
                                ;
010404  012602                  FSECX:  MOV     (SP)+,R2        ;RESTORE R2
010406  000261                          SEC
010410  000207                          RTS     PC
```

```
1874                                    ;
1875                                    ;
1876                                    ;       CONVERT AN ASCII OCTAL VALUE FROM THE COMMAND LINE INTO BINARY
1877                                    ;       LEGAL STRINGS CONTAIN FROM 1 TO 6 CHARACTERS
1878                                    ;
1879                                    ;       INPUT:
1880                                    ;       R0 = NUMBER OF CHARACTERS IN ASCII STRING
1881                                    ;       R1 -> STRING
1882                                    ;
1883                                    ;       OUTPUT:
1884                                    ;       THE FIELD 'BINWD' CONTAINS THE CONVERTED VALUE
1885                                    ;
1886                                    ;
1887 010412                            PACK:
1888 010412  022700  000006                    CMP     #6,R0           ;UPPER LIMIT ON OCTAL DIGITS
1889 010416  001014                            BNE     PSECX           ;ERROR EXIT
1890 010420  010146                            MOV     R1,-(SP)        ;SAVE STRING POINTER TEMPORARILY
1891 010422  060016                            ADD     R0,(SP)         ;ADD CHAR COUNT
1892 010424  010100                            MOV     R1,R0           ;GET STRING ADDR INTO R0 FOR SUBRTN
1893 010426  004767  000000G                   JSR     PC,$COTB        ;CONVERT ASCII OCTAL
1894 010432  005300                            DEC     R0              ;SUBRTN PUSHES R0 1 TOO FAR
1895 010434  020026                            CMP     R0,(SP)+        ;FULL STRING CONVERTED
1896 010436  001004                            BNE     PSECX           ;NO, ERROR
1897 010440  010167  167352                    MOV     R1,BINWD        ;SAVE CONVERTED VALUE
1898                                    ;
1899 010444  000241                    PCLCX:  CLC
1900 010446  000207                            RTS     PC
1901 010450  000261                    PSECX:  SEC
1902 010452  000207                            RTS     PC
1903                                    ;
1904                                    ;
1905                                    ;
1906                                    ;       CONVERT A VALUE FROM BINARY TO PRINTABLE FORM.
1907                                    ;       R1 = WORD TO BE CONVERTED
1908                                    ;       R5 -> PRINT LINE
1909                                    ;
1910                                    ;
1911 010454                            UNPK:
1912 010454                                    SAVE    R0,R1,R2
     010454  010046                            MOV     R0,-(SP)
     010456  010146                            MOV     R1,-(SP)
     010460  010246                            MOV     R2,-(SP)
1913                                    ;
1914 010462  010500                            MOV     R5,R0           ;PREPARE TO CALL CONVERT SUBRTN
1915 010464  012702  000001                    MOV     #1,R2           ;KEEP LEADING ZEROS
1916 010470  004767  000000G                   JSR     PC,$CBOMG       ;CONVERT TO OCTAL ASCII
1917 010474  010005                            MOV     R0,R5           ;RESTORE PRINT LINE POINTER
1918 010476  005205                            INC     R5              ;AND BUMP IT
1919                                    ;
1920 010500                                    RESTOR  R0,R1,R2
     010500  012602                            MOV     (SP)+,R2
     010502  012601                            MOV     (SP)+,R1
     010504  012600                            MOV     (SP)+,R0
1921 010506  000207                            RTS     PC
```

```
1923                                    ;
1924                                    ;
1925                                    ;       CONVERT NUMBER OF PASSES TO ASCII DECIMAL
1926                                    ;       INCLUDE DOUBLE WORD CONVERSION (BINARY TO ASCII DECIMAL)
1927                                    ;
1928                                    ;       INPUT:  R5 -> PRINT LINE
1929                                    ;               FIELD 'PASS' = LOW ORDER PASS COUNT
1930                                    ;               FIELD 'PASSH' =HIGH ORDER PASS COUNT
1931                                    ;
1932                                    ;       OUTPUT: R5 -> NEXT PRINT LINE POSITION AFTER CONVERTED VALUE
1933                                    ;
1934                                    ;
1935 010510                            PASSC:
1936 010510  005767  167444                    TST     PASSH           ;IS PASS NUMBER A DOUBLE WORD
1937 010514  001011                            BNE     DOUBLE          ;YES, CONVERT DOUBLE WORD
1938 010516  010500                            MOV     R5,R0           ;PREPARE TO CALL CONVERSION RTN
1939 010520  016701  167436                    MOV     PASS,R1         ;NUMBER TO BE CONVERTED
1940 010524  012702  000001                    MOV     #1,R2           ;SUPPLY LEADING ZEROS
1941 010530  004767  000000G                   JSR     PC,$CBDSG
1942 010534  010005                            MOV     R0,R5           ;RESTORE PRINT LINE POINTER
1943 010536  000453                            BR      PASSX           ;EXIT
1944                                    ;
1945 010540  016701  167414            DOUBLE: MOV     PASSH,R1        ;LOAD HIGH WORD
1946 010544  016702  167412                    MOV     PASS,R2         ;LOAD LOW WORD
1947 010550  010346                            MOV     R3,-(SP)
1948 010552  010446                            MOV     R4,-(SP)
1949                                    ;
1950 010554  012703  000012                    MOV     #10.,R3         ;R3 = LOOP COUNTER
1951 010560  012704  010670'                   MOV     #ASCNST,R4      ;R4->CONVERSION STORAGE AREA
1952 010564  112724  000060            1$:     MOVB    #60,(R4)+       ;CLEAR STORAGE AREA TO 0'S
1953 010570  005303                            DEC     R3
1954 010572  001374                            BNE     1$
1955                                    ;
1956 010574  012704  010702'                   MOV     #ASCNST+12,R4   ;R4->LAST BYTE OF STORAGE
1957 010600  012703  000012                    MOV     #12,R3
1958 010604  012700  000012            2$:     MOV     #10.,R0
1959 010610                                    CALL    $DDIV
     010610  004767  000000G                   JSR     PC,$DDIV
1960 010614  062700  000060                    ADD     #60,R0          ;ASCII NO. (REMAINDER+60 OCTAL)
1961 010620  110044                            MOVB    R0,-(R4)
1962 010622  005303                            DEC     R3
1963 010624  001367                            BNE     2$
1964                                    ;
1965 010626  012703  000012                    MOV     #12,R3
1966 010632  122724  000060            3$:     CMPB    #60,(R4)+       ;REMOVE LEADING ZEROS
1967 010636  001003                            BNE     4$
1968 010640  005303                            DEC     R3
1969 010642  001373                            BNE     3$
1970 010644  005203                            INC     R3              ;IF ALL 0'S, THE LAST ONE IS OK
1971                                    ;
1972 010646  005304                    4$:     DEC     R4
1973 010650  010401                            MOV     R4,R1           ;R1-> RESULT
1974 010652  010302                            MOV     R3,R2           ;R2 = LENGTH OF RESULT
1975                                    ;
1976 010654  112125                    5$:     MOVB    (R1)+,(R5)+     ;MOVE CONVERTED VALUE TO PRINT LINE
1977 010656  305302                            DEC     R2
1978 010660  001375                            BNE     5$
```

```
1979                                    ;
1980 010662  012604                             MOV     (SP)+,R4
1981 010664  012603                             MOV     (SP)+,R3
1982 010666  000207                     PASSX:  RTS     PC              ;RETURN
1983                                    ;
1984                                    ;
1985 010670                             ASCNST: .BLKW   5               ;CONVERSION STORAGE AREA
```

```
1987                                    ;
1988                                    ;
1989                                    ;       WRITE A PRINT LINE TO TT0
1990                                    ;
1991                                    ;
1992 010702                             CONSOL:
1993 010702  012700  000120                    MOV     #80.,R0         ;PRINT BUFFER BYTE COUNT
1994 010706  012701  001603'                   MOV     #PRINT+78.,R1   ;POINT PAST END OF BUFFER
1995 010712  122741  000040            1$:     CMPB    #40.,-(R1)      ;LOOK FOR A NON-BLANK
1996 010716  001003                            BNE     2$              ;OK, WRITE LINE
1997 010720  005300                            DEC     R0              ;DEC CHAR COUNT
1998 010722  001373                            BNE     1$
1999 010724  000436                            BR      ABEND2          ;NO NON-BLANKS?
2000                                    ;
2001 010726                             2$:     QIOW$S  #IO,WVB,#LUN,TT,#EFN,1,,#STAT,,<#PRINT-2,R0>,ABEND2
     010726  005046                             CLR     -(SP)
     010730  005046                             CLR     -(SP)
     010732  005046                             CLR     -(SP)
     010734  005046                             CLR     -(SP)
     010736  010046                             MOV     R0,-(SP)
     010740  012746  001463'                   MOV     #PRINT-2,-(SP)
     010744  005046                             CLR     -(SP)
     010746  012746  000006'                   MOV     #STAT,-(SP)
     010752  005046                             CLR     -(SP)
     010754  112716  000001                    MOVB    #EFN.1,(SP)
     010760  012746  000001                    MOV     #LUN.TT,-(SP)
     010764  012746  000000G                   MOV     #IO.WVB,-(SP)
     010770  012746                            MOV     (PC)+,-(SP)
     010772     003     014                    .BYTE   3,12.
     010774  104377                            EMT     ^O<377>
     010776  103002                            BCC     .+6
     011000  004767  000016                    JSR     PC,ABEND2
2002                                    ;
2003                                    ;
2004 011004  012701  001465'                   MOV     #PRINT,R1       ;POINT TO STRING
2005 011010  112721  000040            4$:     MOVB    #40.,(R1)+      ;CLEAR LINE TO BLANKS
2006 011014  005300                            DEC     R0              ;DEC LOOP COUNT
2007 011016  001374                            BNE     4$
2008 011020  000207                            RTS     PC
2009                                    ;
2010 011022                             ABEND2: ABRT$S  #MYSELF
     011022  012746  000000'                   MOV     #MYSELF,-(SP)
     011026  017646  000000                    MOV     @(SP),-(SP)
     011032  062766  000002  000002            ADD     #2,2(SP)
     011040  017666  000002  000002            MOV     @2(SP),2(SP)
     011046  012746                            MOV     (PC)+,-(SP)
     011050     123     003                    .BYTE   83.,3
     011052  104377                            EMT     ^O<377>
```

```
                                        ;
                                        ;
                                        ;       WRITE TO TT0 AND PROMPT
                                        ;
                                        ;
2017 011054  005267  166732            ENDTST: INC     ERWORD
2018 011060  005267  166726            HLTTST: INC     ERWORD
2019 011064  005267  166722            STOP:   INC     ERWORD
2020 011070  005267  166716            BEGTST: INC     ERWORD
2021 011074  005267  166712            OUT1:   INC     ERWORD
2022 011100  005267  166706            ERR12:  INC     ERWORD
2023 011104  005267  166702            ERR11:  INC     ERWORD
2024 011110  005267  166676            ERR10:  INC     ERWORD
2025 011114  005267  166672            ERR9:   INC     ERWORD
2026 011120  005267  166666            ERR8:   INC     ERWORD
2027 011124  005267  166662            ERR7:   INC     ERWORD
2028 011130  005267  166656            ERR6:   INC     ERWORD
2029 011134  005267  166652            ERR5:   INC     ERWORD
2030 011140  005267  166646            ERR4:   INC     ERWORD
2031 011144  005267  166642            ERR3:   INC     ERWORD
2032 011150  005267  166636            ERR2:   INC     ERWORD
2033                                    ;
2034 011154                             NESTOP:
2035 011154  005267  166632            EROPT:  INC     ERWORD
2036 011160  005267  166626            LPTST:  INC     ERWORD
2037 011164  005267  166622            USRTST: INC     ERWORD
2038 011170  005267  166616            SELTST: INC     ERWORD
2039 011174  005267  166612            PMPTT:  INC     ERWORD
2040 011200  005267  166606            PMPTO:  INC     ERWORD
2041 011204  005267  166602            PMPTI:  INC     ERWORD
2042 011210  005267  166576            PMPTC:  INC     ERWORD
2043 011214  005267  166572            PMPTB:  INC     ERWORD
2044 011220  005267  166566            PMPTA:  INC     ERWORD
2045 011224  005267  166562            MEMSEL: INC     ERWORD
2046 011230  005267  166556            ALLSEL: INC     ERWORD
2047         000014                     NEST    =       <.-NESTOP>/4
2048                                    ;
2049                                    ;       USE THE INDEX ERWORD TO COUNT UP FROM THE BOTTOM OF THE
2050                                    ;       MESSAGE TABLE, FIND THE END OF THE MESSAGE FIRST, THEN
2051                                    ;       THE BEGINNING, THEN GET THE LENGTH.
2052                                    ;
2053 011234  010246                            MOV     R2,-(SP)        ;SAVE R2
2054 011236  016702  166550                    MOV     ERWORD,R2       ;LOAD LOOP COUNT
2055 011242  012701  003271'                   MOV     #ASCIZ,R1       ;POINT TO END OF MESSAGE TABLE
2056 011246  105741                    1$:     TSTB    -(R1)           ;LOOK FOR END OF MESSAGE
2057 011250  001376                            BNE     1$
2058 011252  005302                            DEC     R2              ;LOOP COUNT
2059 011254  001374                            BNE     1$              ;BACK UP ANOTHER MESSAGE
2060 011256  010100                            MOV     R1,R0           ;SAVE POINTER TO END OF MESSAGE
2061 011260  105741                    2$:     TSTB    -(R1)           ;BACK UP TO BEGINNING OF MESSAGE
2062 011262  001376                            BNE     2$
2063 011264  005201                            INC     R1              ;BUMP TO FIRST CHAR OF MESSAGE
2064 011266  160100                            SUB     R1,R0           ;R0 NOW = MESSAGE LENGTH
2065 011270  012602                            MOV     (SP)+,R2
2066                                    ;
2067 011272                                    QIOW$S  #IO.WVB,#LUN.TT,#EFN.1,,#STAT,,<R1,R0>,ABEND
     011272  005046                            CLR     -(SP)
```

```
        011274  005046                          CLR     -(SP)
        011276  005046                          CLR     -(SP)
        011300  005046                          CLR     -(SP)
        011302  010046                          MOV     R0,-(SP)
        011304  010146                          MOV     R1,-(SP)
        011306  005046                          CLR     -(SP)
        011310  012746  000006'                 MOV     #STAT,-(SP)
        011314  005046                          CLR     -(SP)
        011316  112716  000001                 MOVB    #EFN.1,(SP)
        011322  012746  000001                 MOV     #LUN.TT,-(SP)
        011326  012746  000000G                 MOV     #IO.WVB,-(SP)
        011332  012746                          MOV     (PC)+,-(SP)
        011334      003     014                 .BYTE   3,12.
        011336  104377                          EMT     ^O<377>
        011340  103002                          BCC     .+6
        011342  004767  000106                  JSR     PC,ABEND
  2068                                  ;
  2069  011346                                  CLEF$S  #EFN.1
        011346  012746  000001                 MOV     #EFN.1,-(SP)
        011352  012746                          MOV     (PC)+,-(SP)
        011354      037     002                 .BYTE   31.,2
        011356  104377                          EMT     ^O<377>
  2070  011360  105767  166422                  TSTB    STAT            ;GOOD RETURN
  2071  011364  003433                          BLE     ABEND           ;NO
  2072                                  ;
  2073                                  ;       ISSUE GCML
  2074                                  ;
  2075  011366  022767  000014  166416          CMP     #NEST,ERWORD    ;PROMPT WITH MESSAGE
  2076  011374  002424                          BLT     TTX             ;NO, JUST EXIT
  2077                                  ;
  2078  011376  012700  000032'                MOV     #GCMBUF,R0      ;POINT TO GCML BUFFER
  2079  011402  012701  000051                 MOV     #41.,R1         ;NUMBER OF WORDS
  2080  011406  005020                  3$:    CLR     (R0)+           ;CLEAR BUFFER
  2081  011410  005301                          DEC     R1
  2082  011412  001375                          BNE     3$
  2083                                  ;
  2084  011414                                  GCML$   #GCMBLK
        011414  012700  003322'                MOV     #GCMBLK,R0
        011420  005060  000142                  CLR     G.PSDS(R0)
        011424  004767  000000G                 JSR     PC,.GCML1
  2085  011430  103411                          BCS     ABEND
  2086  011432  016867  000146  166514         MOV     G.CMLD(R0),GCMLEN       ;SAVE LENGTH
  2087  011440  012767  000032' 166510         MOV     #GCMBUF,GCMPNT  ;INITIALIZE COMMAND BUFFER @POINTER
  2088  011446  005067  166340          TTX:   CLR     ERWORD          ;CLEAR ERROR NUMBER INDICATOR
  2089  011452  000207                          RTS     PC              ;AND RETURN
  2090                                  ;
  2091  011454                          ABEND:  ABRT$S  #MYSELF
        011454  012746  000000'                MOV     #MYSELF,-(SP)
        011460  017646  000000                  MOV     @(SP),-(SP)
        011464  062766  000002  000002          ADD     #2,2(SP)
        011472  017666  000002  000002          MOV     @2(SP),2(SP)
        011500  012746                          MOV     (PC)+,-(SP)
        011502      123     003                 .BYTE   83.,3
        011504  104377                          EMT     ^O<377>
  2092                                  ;
  2093          003630'                         .END    START
```

```
A      = 000001
ABEND    011454R
ABEND2   011022R
ALL      003634R
ALLSEL   011230R
ALLSTR   000401R
ALLTST=  000010
AMSG     001355R
ASCIZ    003271R
ASCNST   010670R
AST      007206R
ASTWRD   000004R
ASWRK    001316R
B      = 000002
BASE     000024R
BEGTST   011070R
BINWD    000016R
BITVAL=  000000
BIT0   = 000001
BIT1   = 000002
BIT10  = 002000
BIT11  = 004000
BIT12  = 010000
BIT13  = 020000
BIT14  = 040000
BIT15  = 100000
BIT2   = 000004
BIT3   = 000010
BIT4   = 000020
BIT5   = 000040
BIT6   = 000100
BIT7   = 000200
BIT8   = 000400
BIT9   = 001000
BYTE0  = 000000
BYTE1  = 000001
BYTE2  = 000002
BYTE3  = 000003
BYTE4  = 000004
BYTE5  = 000005
BYTE6  = 000006
BYTE7  = 000007
BYTE8  = 000010
BYTE9  = 000011
BYTVAL=  000012
C      = 000004
CAST     007460R
CF.COT=  000041
CF.DGN=  000046
CF.DHR=  000042
CF.DMC=  000047
CF.HBR=  000045
CF.HRL=  000044
CF.UPD=  000043
CHECK    004616R
CHECK0   005000R
CKDATA   001274RG

CK2      001276RG
CK3      001300RG
CMILUN=  000002
CNUM   = 000006
CONSOL   010702R
CTBL     000430R
CURLIM   001150R
DBSLEN=  000116
DG.ERR=  001000
DG.SDF=  002000
DG.TDF=  004000
DOUBLE   010540R
EAST     007564R
EFN.1  = 000001
EMSG     001366R
ENDLIM   004240R
ENDLN  = 000014
ENDOF    003306R
ENDTST   011054R
ERLIM    000014R
EROPT    011154R
ERPRMT   005160R
ERRADD   001304RG
ERRCT    001306RG
ERROR  = 000040
ERRORX   007170R
ERR10    011110R
ERR11    011104R
ERR12    011100R
ERR2     011150R
ERR3     011144R
ERR4     011140R
ERR5     011134R
ERR6     011130R
ERR7     011124R
ERR8     011120R
ERR9     011114R
ERWORD   000012R
ERW1     001310RG
ERW2     001312RG
ERW3     001314RG
FD.CCL=  ****** GX
FD.FID   000000
FD.FNB   000006
FD.FVR   000004
FD.LEN   000010
FD.REC=  ****** GX
FD.TTY=  ****** GX
FIND     010274R
FMSG     001335R
FNIN1    010216R
FNMTCH   010252R
FNOUT1   010210R
FNOUT2   010230R
FN.ACK   000016
FN.FSA   000000
FN.FSB   000002

    FN.FSC   000004
    FN.FSD   000020
    FN.MHR   000010
    FN.NMB   000022
    FN.QLS   000006
    FN.RDC   000014
    FN.UPD   000012
    FSECX    010404R
    FTBL     001417R
    F.ACTL=  000076
    F.ALOC=  000040
    F.BBFS=  000062
    F.BDB  = 000070
    F.BGBC=  000057
    F.BKDN=  000026
    F.BKDS=  000020
    F.BKEF=  000050
    F.BKP1=  000051
    F.BKST=  000024
    F.BKVB=  000064
    F.CHR  = 000075
    F.CNTG=  000034
    F.DFNB=  000046
    F.DSPT=  000044
    F.DVNM=  000134
    F.EFBK=  000010
    F.EFN  = 000050
    F.EOBB=  000032
    F.ERR  = 000052
    F.FACC=  000043
    F.FFBY=  000014
    F.FNAM=  000110
    F.FNB  = 000102
    F.FTYP=  000116
    F.FVER=  000120
    F.HIBK=  000004
    F.LUN  = 000042
    F.MBCT=  000054
    F.MBC1=  000055
    F.MBFG=  000056
    F.NRBD=  000024
    F.NREC=  000030
003 F.OVBS=  000030
003 F.RACC=  000016
003 F.RATT=  000001
003 F.RCNM=  000034
    F.RCTL=  000017
    F.RSIZ=  000002
    F.RTYP=  000000
    F.SEQN=  000100
    F.SPDV=  000072
    F.SPUN=  000074
    F.STBK=  000036
    F.UNIT=  000136
004 F.URBD=  000020
004 F.VBN  = 000064
004 F.VBSZ=  000060

004 GCMBLK   003322R
004 GCMBUF   000032R
004 GCMLEN   000154R
004 GCMPNT   000156R
004 GE.BIF=  177775
004 GE.CLO=  000004
004 GE.COM=  000001
    GE.CON=  000020
    GE.EOF=  177766
    GE.IND=  000002
    GE.IOR=  177777
    GE.LC  = 000010
    GE.MDE=  177774
    GE.OPR=  177776
    GE.RBG=  177730
    GE.SIZ=  000040
    G.CMLD=  000146
    G.DPRM=  000160
    G.ERR  = 000140
    G.ISIZ=  000020
    G.LPDL=  000060
    G.MODE=  000141
    G.PSDS=  000142
    G.SIZE=  000224
    HALT   = 000020
    HLTTST   011060R
    I      = 000010
    IO.ATA=  ****** GX
    IO.WVB=  ****** GX
    JMPMT    005614R
    LASTJ    006036R
    LASTJ1   006040R
    LIMITS   010002R
    LIMNUM=  000014
    LIMREF   000700R
    LIMX     010164R
    LIMX2    010200R
    LOOP   = 000001
    LOOPCT   000164R
    LOWER    000022R
    LPRMPT   005032R
    LPTST    011160R
    LUN.TT=  000001
    MEM    = 000006
    MEMERR   006540RG
    MEMSEL   011224R
    MEMTOP   004012R
    MOVE     003706R
    MS.DGN=  010000
    MTCNT    000172R
    MTJUMP   006530R
    MTMAIN   005400R
    MTPNT    000170R
    MTREF    000460R
    MTRTN    003764R
    MTSET    005304R
    MTSUB    000730R

MT10     000026R
MYSELF   000000R
N      = 000012
NEST   = 000014
NESTOP   011154R
N.BFAC=  000004
N.BHGH=  000006
N.BTCH=  000004
N.BUFB=  004000
N.BUFW=  002000
N.DID  = 000024
N.DVNM=  000032
N.FID  = 000000
N.FNAM=  000006
N.FOS  = 000764
N.FTYP=  000014
N.FVER=  000016
N.NEXT=  000022
N.QURY=  000031
N.STAT=  000020
N.SUNT=  000002
N.UNIT=  000034
O      = 000020
OUT1     011074R
PACK     010412R
PAR$$$=  000027
PASS     000162R
PASSC    010510R
PASSH    000160R
PASSX    010666R
PAST     007600R
PCLCX    010444R
PLIM     004064R
PMPTA    011220R
PMPTB    011214R
PMPTC    011210R
PMPTI    011204R
PMPTO    011200R
PMPTT    011174R
PMPT10   004532R
PMSG     001330R
PMSG2    003272R
PM2LN  = 000014
PREADD   001302RG
PRINT    001465R
PSECX    010450R
REST     007652R
RMSG     001400R
SCAN     010202R
SELECT   000166R
SELOOP   004314R
SELTST   011170R
SERR     004356R
SETTBL   000174R
SR.ARE   000114     002
SR.ARS   000106     002
SR.DAY   000010     002
```

```
SR.DLT  000014      002 S.FNAM= 000006      T$BT   = 000020      T$1CLK= 000400      UNMSG   001412R
SR.ECB  000047      002 S.FNB  = 000036      T$BTAR= 000030      T$8BEN= 000020      UNPK    010454R
SR.ECH  000046      002 S.FNBW= 000017      T$BTD  = 002000      Ti      006064R     UPPER   000020R
SR.ECL  000050      002 S.FNTY= 000004      T$CD   = 000100      TiADDR  001214R     USRTST  011164R
SR.FIB  000012      002 S.FTYP= 000002      T$CLK  = 002000      TiF    = ****** GX  WAST    007306R
SR.GRE  000100      002 S.NFEN= 000020      T$DISK= 000200      TiI    = ****** GX  WORD0  = 000000
SR.GRS  000072      002 T      = 000040      T$DRD  = 000004      TiO    = ****** GX  WORD1  = 000002
SR.LEN  000122      002 TAST    007502R      T$EMEM= 010000      TiT    = ****** GX  WORD2  = 000004
SR.LIN  000066      002 TCDADD  001244R      T$FSAA= 000000      Ti0     006274R     WORD3  = 000006
SR.LIP  000062      002 TCUADD  001260R      T$FSAB= 000004      Ti1     006300R     WORD4  = 000010
SR.MON  000006      002 TD$CTR= 176370       T$FSAC= 000014      Ti2     006424R     WORD5  = 000012
SR.NDC  000042      002 TD$CTW= 176360       T$FSB2= 000010      Ti2FD  = ****** GX  WORD6  = 000014
SR.NDS  000036      002 TD$INL= 004000       T$IB   = 000026      Ti2FU  = ****** GX  WORD7  = 000016
SR.NIN  000030      002 TD$MEM= 000270       T$IBAR= 000024      Ti2ID  = ****** GX  WORD8  = 000020
SR.NIP  000022      002 TD$OAR= 176344       T$IBE  = 020000      Ti2IU  = ****** GX  WORD9  = 000022
SR.SDB  000032      002 TD$OTR= 176346       T$IBF  = 040000      Ti2OD  = ****** GX  WRDVAL= 000024
SR.SRC  000002      002 TD$ORD= 000274       T$ICD  = 000040      Ti2OU  = ****** GX  ZREL     004402R
SR.SUN  000000      002 TD$RST= 176366       T$MODE= 004000      Ti2TD  = ****** GX  $CBDSG= ****** GX
SR.TWS  000056      002 TD$SW  = 176376       T$OB   = 000036      Ti2TU  = ****** GX  $CBOMG= ****** GX
SR.WSL  000052      002 TD$TAR= 176372       T$OBE  = 004000      T2      006074R     $CDTB  = ****** GX
SR.YR   000004      002 TD$TAW= 176362       T$OBF  = 010000      T3      006110R     $COTB  = ****** GX
SR.1IN  000024      002 TD$TDR= 176374       T$OBRA= 000034      T4      006126R     $DDIV  = ****** GX
SR.1IP  000016      002 TD$TDW= 176364       T$OBWA= 000032      T5      006144R     $DIV   = ****** GX
STADDR  001200R          TESTS  = 000014      T$OUTA= 100000      T6      006174R     $MUL   = ****** GX
START   003630R          TEST10= 000004       T$RBDO= 000200      T6CHK   004574R     $$$    = 003502R
STAT    000006R          TEST6  = 000002      T$RNB  = 000040      T6FACT  000030R     $$$ARG= 000002
STOP    011064R          TMSG    001323R      T$RSET= 040000      T7      006200R     $$$T1  = 000067
STRLEN= 000027           TROCT   000202R      T$SC   = 000022      T7ADDR  001230R     $$$T2  = 000027
STUFF  = ****** GX       TRT     004344R      T$SCLK= 020000      T7F    = ****** GX  .FSRCB= ****** G
STUFI  = ****** GX       TTX     011446R      T$SEG1= 000000      T7I    = ****** GX  .GEML1= ****** G
STUFO  = ****** GX       T$AD   = 000020      T$SEG2= 000001      T7O    = ****** GX  ...PC1= 003322R
STUFT  = ****** GX       T$BA   = 000002      T$SEG3= 000002      T7T    = ****** GX  ...PC2= 003524R
S.BFHD= 000020           T$BD   = 000010      T$SO   = 000001      T8      006210R     ...PC3= 003322R
S.FATT= 000016           T$BSO  = 100000      T$UBUS= 100000      T9      006240R     ...TPC= 000020
S.FDB  = 000140

. ABS.  000000    000
        011506    001
SRCOFF  000122    002
FDSCOF  000010    003
FNOFFS  000022    004
$$FSR1  001020    005
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  7110 WORDS  (28 PAGES)
DYNAMIC MEMORY:  8084 WORDS  ( 31 PAGES)
ELAPSED TIME:  00:01:25
TMT,TMT/-SP=C20,1JP,C,TMT
```

```
                                        .TITLE  TTEST
000000                                  .PSECT  TTEST
                                        .LIST   MEB
                                ;
                                ;
                                ;
                                ;       TERM DETECTOR MEMORY DIAGNOSTICS
                                ;
                                ;
                                ;
                                ;       TEST ROUTINES FOR FSA'S
                                ;
                                ;
                                ;       THIS ROUTINE WRITES AND READS THE TEST PATTERN CONTAINED
                                ;       IN 'CKDATA'
                                ;
000000                          STUFF::
000000  016667  000002  000000G        MOV     2(SP),PREADD
000006  016767  000000G 176362  1$:     MOV     PREADD,TD$TAW
000014  004767  000474                  JSR     PC,WRITE
000020  005267  000000G                 INC     PREADD
000024  026667  000004  000000G        CMP     4(SP),PREADD    ;CHECK FOR UPPER LIMITS
000032  103365                          BHIS    1$
                                ;
000034  016667  000002  000000G        MOV     2(SP),PREADD
000042  016667  000002  176362         MOV     2(SP),TD$TAW
000050  004767  000464          2$:     JSR     PC,READ
000054  005267  000000G                 INC     PREADD
000060  026667  000004  000000G        CMP     4(SP),PREADD    ;CHECK FOR UPPER LIMITS
000066  103370                          BHIS    2$
000070  000207                          RTS     PC
                                ;
                                ;
                                ;       TEST 01
                                ;       WRITE ADDRESS INTO ITSELF
                                ;
                                ;
000072                          T1F::
000072  016667  000002  000000G        MOV     2(SP),CKDATA    ;ADDRESS IS TEST PATTERN
000100  016667  000002  000000G        MOV     2(SP),PREADD
000106  016767  000000G 176362  1$:     MOV     PREADD,TD$TAW
000114  004767  000374                  JSR     PC,WRITE
000120  005267  000000G                 INC     CKDATA
000124  005267  000000G                 INC     PREADD
000130  026667  000004  000000G        CMP     4(SP),PREADD    ;CHECK FOR UPPER LIMITS
000136  103363                          BHIS    1$
                                ;
000140  016667  000002  000000G        MOV     2(SP),CKDATA    ;ADDRESS IS TEST PATTERN
000146  016667  000002  000000G        MOV     2(SP),PREADD
000154  016667  000002  176362         MOV     2(SP),TD$TAW
000162  004767  000352          2$:     JSR     PC,READ
000166  005267  000000G                 INC     CKDATA
000172  005267  000000G                 INC     PREADD
000176  026667  000004  000000G        CMP     4(SP),PREADD    ;CHECK FOR UPPER LIMITS
000204  103366                          BHIS    2$
000206  000207                          RTS     PC
                                ;
```

```
                                ;
                                ;       TEST·06
                                ;       MEMORY·CROSS-TALK·TEST
                                ;
                                ;
000210                          T6F::
                                ;       MOV·    #10.,R2·        ;SET·LOOP·COUNT·
                                ;       MOV·    #-1,CKDATA·     ;SET·TEST·PATTERN·TO·ALL·1'S
                                ;10$:   MOV·    2(SP),PREADD·   ;SET·UP·START·ADDRESS·
                                ;1$:    MOV·    PREADD,TD$TAW·  ;MOVE·ADDR·TO·TRANSFER·REG·
                                ;       JSR·    PC,WRITE·       ;WRITE·'CKDATA'·TO·3·SEGS·
                                ;       ADD·    #2,PREADD·      ;SKIP·AN·ADDRESS·
                                ;       CMP·    4(SP),PREADD·   ;FINISHED·
                                ;       BHIS·   1$              ;NO,·CONTINUE·
                                ;       DEC·    R2·             ;SUB·FROM·LOOP·COUNT·
                                ;       BNE·    10$
                                ;
                                ;       CLR·    CKDATA·         ;SET·TEST·PATTERN·TO·ZERO·
                                ;       MOV·    2(SP),PREADD·   ;RESET·START·ADDRESS·
                                ;       INC·    PREADD·         ;START·AT·FIRST·ADDR·+·1
                                ;2$:    MOV·    PREADD,TD$TAW·  ;MOVE·START·ADDRESS·TO·TRANS.·REG·
                                ;       JSR·    PC,READ·        ;READ·AND·COMPARE·
                                ;       ADD·    #2,PREADD·      ;BUMP·ADDR·
                                ;       CMP·    4(SP),PREADD·   ;FINISHED·
                                ;       BHIS·   2$              ;NO·
000210  000207                  ;       RTS·    PC·
                                ;
                                ;
                                ;       TEST·07
                                ;       WRITE/READ·ALL·ADDRESSES·WITH·ADDRESS·COMPLEMENT·
                                ;
                                ;
000212·                         T7F::
000212·  016602·  000002·               MOV·    2(SP),R2·       ;LOAD·ADDRESS·
000216·  016667  000002·  000000G·      MOV·    2(SP),PREADD·   ;SET·UP·START·ADDRESS·
000224  005102·                 1$:     COM·    R2·             ;GET·ADDRESS·COMPLEMENT·
000226·  010267  000000G·               MOV·    R2,CKDATA·      ;SET·TEST·PATTERN·
000232·  016767  000000G·  176362·      MOV·    PREADD,TD$TAW·  ;MOVE·ADDR·TO·TRANSFER·REG·
000240  004767  000250                  JSR·    PC,WRITE·       ;WRITE·'CKDATA'·TO·3·SEGS·
000244  005267  000000G·               INC·    PREADD·         ;BUMP·TO·NEXT·ADDRESS·
000250  016702·  000000G·               MOV·    PREADD,R2·      ;SET·R2·TO·NEXT·ADDRESS·
000254  026667  000004  000000G·       CMP·    4(SP),PREADD·   ;FINISHED·
000262·  103360                         BHIS·   1$              ;NO,·CONTINUE·
                                ;
000264  016667  000002  176362·       MOV·    2(SP),TD$TAW·   ;MOVE·START·ADDRESS·TO·TRANS·REG·
000272·  016602·  000002·               MOV·    2(SP),R2·       ;SET·R2·=·ADDRESS
000276  016667  000002  000000G·       MOV·    2(SP),PREADD·   ;RESET·START·ADDRESS·
000304  005102·                 2$:     COM·    R2·             ;GET·ADDRESS·COMPLEMENT·
000306  010267  000000G·               MOV·    R2,CKDATA·      ;SET·TEST·PATTERN·
000312·  004767  000222·                JSR·    PC,READ·        ;READ·AND·COMPARE·
000316  005267  000000G·               INC·    PREADD·         ;BUMP·ADDR·
000322·  016702·  000000G·               MOV·    PREADD,R2·      ;SET·R2·=·NEXT·ADDRESS
000326  026667  000004  000000G·       CMP·    4(SP),PREADD·   ;FINISHED·
000334  103363                         BHIS·   2$              ;NO·
000336  000207                         RTS·    PC·
                                ;
                                ;
```

```
                                        ;       TEST 12
                                        ;       ADDRESSING TEST
                                        ;
                                        ;
000340                          T12FD::
000340  016667  000002  000000G      MOV     2(SP),PREADD    ;GET START ADDRESS
000346  016767  000000G 000000G 1$:   MOV     CK2,CKDATA      ;GET READ PATTERN
000354  016767  000000G 176362      MOV     PREADD,TD$TAW
000362  004767  000152              JSR     PC,READ
000366  016767  000000G 000000G      MOV     CK3,CKDATA      ;GET WRITE PATTERN
000374  016767  000000G 176362      MOV     PREADD,TD$TAW
000402  004767  000106              JSR     PC,WRITE
000406  062767  000001  000000G      ADD     #1,PREADD
000414  026667  000004  000000G      CMP     4(SP),PREADD
000422  103351                      BHIS    1$
000424  000207                      RTS     PC
                                        ;
000426                          T12FU::
000426  016667  000004  000000G      MOV     4(SP),PREADD    ;START ADDRESS = UPPER LIMIT
000434  016767  000000G 000000G 1$:   MOV     CK2,CKDATA      ;READ PATTERN
000442  016767  000000G 176362      MOV     PREADD,TD$TAW
000450  004767  000064              JSR     PC,READ
000454  016767  000000G 000000G      MOV     CK3,CKDATA      ;WRITE PATTERN
000462  016767  000000G 176362      MOV     PREADD,TD$TAW
000470  004767  000020              JSR     PC,WRITE
000474  162767  000001  000000G      SUB     #1,PREADD
000502  026667  000002  000000G      CMP     2(SP),PREADD
000510  003751                      BLE     1$
000512  000207                      RTS     PC
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       WRITE SUBROUTINE...
                                        ;
000514  016767  000000G 176364  WRITE: MOV     CKDATA,TD$TDW
000522  016767  000000G 176364      MOV     CKDATA,TD$TDW
000530  016767  000000G 176364      MOV     CKDATA,TD$TDW
000536  000207                      RTS     PC
                                        ;
                                        ;
                                        ;       READ SUBROUTINE...
                                        ;
000540                          READ:
000540                              SAVE    R1,R2,R3
000540  010146                      MOV     R1,-(SP)
000542  010246                      MOV     R2,-(SP)
000544  010346                      MOV     R3,-(SP)
                                        ;
000546  016701  176374              MOV     TD$TDR,R1
000552  020167  000000G              CMP     R1,CKDATA
000556  001403                      BEQ     1$
000560  004767  000100              JSR     PC,PRPERR
000564  000424                      BR      READX
000566  016701  176374      1$:     MOV     TD$TDR,R1
000572  020167  000000G              CMP     R1,CKDATA
000576  001403                      BEQ     2$
```

```
169 000600  004767  000060'                 JSR     PC,PRPERR
170 000604  000414                          BR      READX
171 000606  016702  176374          2$:     MOV     TD$TDR,R2
172 000612  016703  000000G                 MOV     CKDATA,R3
173 000616  042702  000777                  BIC     #000777,R2      ;CLEAR 9 BITS
174 000622  042703  000777                  BIC     #000777,R3
175 000626  020203                          CMP     R2,R3
176 000630  001402                          BEQ     READX
177 000632  004767  000026'                 JSR     PC,PRPERR
178                                 ;
179 000636                          READX:  RESTOR  R1,R2,R3
    000636  012603                          MOV     (SP)+,R3
    000640  012602                          MOV     (SP)+,R2
    000642  012601                          MOV     (SP)+,R1
180 000644  000207                          RTS     PC
181                                 ;
182                                 ;
183                                 ;       READ WITHOUT COMPARE
184                                 ;
185                                 ;
186 000646                          READ1:
187 000646  016746  176374                  MOV     TD$TDR,-(SP)
188 000652  016716  176374                  MOV     TD$TDR,(SP)
189 000656  016726  176374                  MOV     TD$TDR,(SP)+
190 000662  000207                          RTS     PC
191                                 ;
192                                 ;
193                                 ;       PRE@ARE TO CALL ERROR ROUTINE
194                                 ;
195 000664                          PRPERR:
196 000664  016767  000000G 000000G         MOV     PREADD,ERRADD
197 000672  016767  000000G 176362          MOV     ERRADD,TD$TAW   ;SET ERR ADDRESS
198 000700  012767  000003  000000G         MOV     #3,ERRCT        ;SEGMENT COUNT
199 000706  016767  176374  000000G         MOV     TD$TDR,ERW1     ;READ SEGMENT 0
200 000714  016767  176374  000000G         MOV     TD$TDR,ERW2     ;READ SEGMENT 1
201 000722  016767  176374  000000G         MOV     TD$TDR,ERW3     ;READ SEGMENT 2
202 000730  042767  000777  000000G         BIC     #000777,ERW3
203                                 ;
204                                 ;       RETRY READ
205                                 ;
206 000736  012746  000005                  MOV     #5,-(SP)        ;NUMBER OF RETRIES
207 000742  016767  000000G 176362  RR1:    MOV     ERRADD,TD$TAW
208 000750  016746  176374                  MOV     TD$TDR,-(SP)    ;READ SEG 1
209 000754  016746  176374                  MOV     TD$TDR,-(SP)    ;SEG 2
210 000760  016746  176374                  MOV     TD$TDR,-(SP)    ;SEG 3
211 000764  042716  000777                  BIC     #000777,(SP)
212 000770  022667  000000G                 CMP     (SP)+,ERW3      ;SAME AS PREVIOUS
213 000774  001012                          BNE     1$              ;NO, ERROR UNPREDICTABLE
214 000776  022667  000000G                 CMP     (SP)+,ERW2      ;SAME AS PREVIOUS
215 001002  001011                          BNE     2$              ;NO
216 001004  022667  000000G                 CMP     (SP)+,ERW1      ;SAME
217 001010  001010                          BNE     3$              ;NO
218 001012  005316                          DEC     (SP)            ;SUB FROM #.RETRIES
219 001014  001352                          BNE     RR1
220 001016  005726                          TST     (SP)+
221 001020  000411                          BR      CALLER          ;CALL ERROR REPORTING ROUTINE
222
```

```
223 001022· 062706  000002          1$:     ADD·    #2,SP·
224 001026  062706  000002          2$:     ADD·    #2,SP·
225 001032· 062706  000002          3$:     ADD·    #2,SP·
226 001036  012767  177777  000000G·        MOV·    #-1,ERRCT·      ;INDICATE·UNPREDICTABLE·RESULTS·
227                                 ;
228 001044  004767  000000G·        CALLER: JSR·    PC,MEMERR·
229 001050  000207                          RTS·    PC·
```

```
                                ;
                                ;
                                ;       INPUT BUFFER TESTS.
                                ;
                                ;
                                ;       WRITE AND READ THE TEST PATTERN.
                                ;
001052                          STUFI::
001052  016667  000002  000000G       MOV     2(SP),PREADD
001060  016701  000000G       1$:     MOV     PREADD,R1
001064  016711  000000G               MOV     CKDATA,(R1)
001070  012767  000000  176376        MOV     #0,TD$SW
001076  062767  000002  000000G       ADD     #2,PREADD
001104  026667  000004  000000G       CMP     4(SP),PREADD
001112  103362                        BHIS    1$
                                ;
001114  016667  000002  000000G       MOV     2(SP),PREADD
001122  004767  000416          2$:   JSR     PC,READI        ;READ AND COMPARE
001126  026667  000004  000000G       CMP     4(SP),PREADD
001134  103372                        BHIS    2$
001136  000207                        RTS     PC
                                ;
                                ;
                                ;       TEST 01
                                ;       WRITE ADDRESS INTO ITSELF
                                ;
                                ;
001140                          T1I::
001140  016667  000002  000000G       MOV     2(SP),PREADD
001144  016767  000000G 000000G 1$:    MOV     PREADD,CKDATA
001154  016701  000000G               MOV     PREADD,R1
001160  016711  000000G               MOV     CKDATA,(R1)
001164  012767  000000  176376        MOV     #0,TD$SW
001172  062767  000002  000000G       ADD     #2,PREADD
001200  026667  000004  000000G       CMP     4(SP),PREADD    ;CHECK FOR UPPER LIMITS
001206  103357                        BHIS    1$
                                ;
001210  016667  000002  000000G       MOV     2(SP),PREADD
001216  016767  000000G 000000G 2$:    MOV     PREADD,CKDATA   ;ADDRESS IS PATTERN
001224  004767  000314                JSR     PC,READI
001230  026667  000004  000000G       CMP     4(SP),PREADD    ;CHECK FOR UPPER LIMITS
001236  103367                        BHIS    2$
001240  000207                        RTS     PC
                                ;
                                ;
                                ;       TEST 06
                                ;       MEMORY CROSS-TALK TEST
                                ;
                                ;
001242                          T6I::
                                ;       MOV     #10.,R2         ;SET LOOP COUNT
                                ;       MOV     #-1,CKDATA      ;SET TEST PATTERN TO ALL 1'S
                                ;10$:   MOV     2(SP),PREADD    ;SET UP START ADDRESS
                                ;1$:    MOV     PREADD,R1       ;LOAD ADDRESS
                                ;       MOV     CKDATA,(R1)
                                ;       MOV     #0,TD$SW
                                ;       ADD     #4,PREADD       ;SKIP AN ADDRESS
```

```
                                        ;       CMP     4(SP),PREADD    ;FINISHED
                                        ;       BHIS    1$              ;NO, CONTINUE
                                        ;       DEC     R2              ;SUB FROM LOOP COUNT
                                        ;       BNE     10$
                                        ;
                                        ;       CLR     CKDATA          ;SET TEST PATTERN TO ZERO
                                        ;       MOV     2(SP),PREADD    ;RESET START ADDRESS
                                        ;       ADD     #2,PREADD
                                        ;2$:    JSR     PC,READI        ;READ AND COMPARE
                                        ;       ADD     #2,PREADD       ;BUMP TO NEXT ADDR (SKIP ONE)
                                        ;       CMP     4(SP),PREADD    ;FINISHED
                                        ;       BHIS    2$              ;NO
000207                                  RTS     PC
                                        ;
                                        ;
                                        ;       TEST 07
                                        ;       WRITE/READ ALL ADDRESSES WITH ADDRESS COMPLEMENT
                                        ;
                                        ;
001244                                  T7I::
016602 000002                           MOV     2(SP),R2        ;SET R2 = ADDRESS
016667 000002 000000G                   MOV     2(SP),PREADD    ;SET UP START ADDRESS
005102                                  1$:     COM     R2              ;GET ADDRESS COMPLEMENT
010267 000000G                          MOV     R2,CKDATA       ;SET TEST PATTERN = ADDRESS COMPLEMENT
016701 000000G                          MOV     PREADD,R1       ;LOAD ADDRESS
016711 000000G                          MOV     CKDATA,(R1)
012767 000000 176376                    MOV     #0,TD$SW
062767 000002 000000G                   ADD     #2,PREADD
016702 000000G                          MOV     PREADD,R2       ;SET R2 = ADDRESS
026667 000004 000000G                   CMP     4(SP),PREADD    ;FINISHED
103355                                  BHIS    1$              ;NO, CONTINUE
                                        ;
016602 000002                           MOV     2(SP),R2        ;SET R2 = ADDRESS
016667 000002 000000G                   MOV     2(SP),PREADD    ;RESET START ADDRESS
005102                                  2$:     COM     R2              ;GET ADDRESS COMPLEMENT
010267 000000G                          MOV     R2,CKDATA       ;SET TEST PATTERN = ADDR COM
004767 000174                           JSR     PC,READI        ;READ AND COMPARE
016702 000000G                          MOV     PREADD,R2       ;SET R2 = NEXT ADDRESS
026667 000004 000000G                   CMP     4(SP),PREADD    ;FINISHED
103365                                  BHIS    2$              ;NO
000207                                  RTS     PC
                                        ;
                                        ;
                                        ;       TEST 12
                                        ;       ADDRESSING TEST
                                        ;
                                        ;
001366                                  T12ID::
016667 000002 000000G                   MOV     2(SP),PREADD    ;GET START ADDRESS
016701 000000G                          1$:     MOV     PREADD,R1       ;LOAD READ ADDRESS
011101                                  MOV     (R1),R1         ;READ DATA
026701 000000G                          CMP     CK2,R1          ;COMPARE AGAINST PATTERN
001402                                  BEQ     .+6
004767 000166                           JSR     PC,PRPER1
                                        ;
016701 000000G                          MOV     PREADD,R1
016711 000000G                          MOV     CK3,(R1)        ;WRITE PATTERN
```

```
 345 001424  012767  000000  176376          MOV·    #0,TD$SW·
 346 001432· 062767  000002  000000G·        ADD·    #2,PREADD·
 347 001440  026667  000004  000000G·        CMP·    4(SP),PREADD·
 348 001446  103352·                         BHIS·   1$
 349 001450  000207                          RTS·    PC·
 350                                 ;
 351 001452·                         T12IU::
 352·001452· 016667  000004  000000G·        MOV·    4(SP),PREADD·    ;START·ADDRESS·=·UPPER·LIMIT
 353 001460  016701  000000G·        1$:     MOV·    PREADD,R1
 354 001464  011101                         MOV·    (R1),R1
 355 001466  012767  000000  176376          MOV·    #0,TD$SW·
 356 001474  026701  000000G·                CMP·    CK2,R1
 357 001500  001402·                         BEQ·    .+6
 358 001502· 004767  000074                  JSR·    PC,PRPER1
 359                                 ;
 360 001506  016701  000000G·                MOV·    PREADD,R1
 361 001512· 016711  000000G·                MOV·    CK3,(R1)         ;WRITE·PATTERN
 362·001516  012767  000000  176376          MOV·    #0,TD$SW·
 363 001524  162767  000002  000000G·        SUB·    #2,PREADD·       ;BACK·UP
 364 001532· 026667  000002  000000G·        CMP·    2(SP),PREADD·
 365 001540  101747                          BLOS·   1$
 366 001542· 000207                          RTS·    PC·
 367                                 ;
 368                                 ;
 369                                 ;       READ·AND·COMPARE·ROUTINE·
 370                                 ;
 371                                 ;
 372·001544  016701  000000G·        READI:  MOV·    PREADD,R1
 373 001550  011101                         MOV·    (R1),R1
 374 001552  012767  000000  176376          MOV·    #0,TD$SW·
 375 001560  026701  000000G·                CMP·    CKDATA,R1
 376 001564  001402·                         BEQ·    .+6
 377 001566  004767  000010                  JSR·    PC,PRPER1
 378 001572  062767  000002  000000G·        ADD·    #2,PREADD·
 379 001600  000207                          RTS·    PC·
 380                                 ;
 381                                 ;
 382·                                ;       INPUT·BUFFER ERROR·PREPARATION·ROUTINE·
 383                                 ;
 384                                 ;
 385 001602·                         PRPER1:
 386 001602·                                 SAVE·   R1,R2,R3
     001602· 010146                          MOV·    R1,-(SP)
     001604  010246                          MOV·    R2,-(SP)
     001606  010346                          MOV·    R3,-(SP)
 387                                 ;
 388 001610  016767  000000G·000000G·        MOV·    PREADD,ERRADD·
 389 001616  012767  000001· 000000G·        MOV·    #1,ERRCT
 390 001624  016701  000000G·                MOV·    PREADD,R1
 391 001630  011167  000000G·                MOV·    (R1),ERWi
 392·                                ;
 393                                 ;       RETRY·
 394                                 ;
 395 001634  012702· 000005                  MOV·    #5,R2·           ;NUMBER·OF·RETRIES
 396 001640  011103                  1$:     MOV·    (R1),R3          ;GET·A·WORD·FROM·MEMORY
 397 001642· 026703  000000G·                CMP·    ERWi,R3
 398 001646  001003                          BNE·    2$
```

```
399 001650  005302                          DEC     R2
400 001652  001372                          BNE     1$
401 001654  000403                          BR      3$
402                                 ;
403 001656  012767  177777  000000G 2$:     MOV     #-1,ERRCT       ;INDICATE UNPREDICTABLE RESULTS
404 001664  004767  000000G         3$:     JSR     PC,MEMERR
405                                 ;
406 001670                                  RESTOR  R1,R2,R3
    001670  012603                          MOV     (SP)+,R3
    001672  012602                          MOV     (SP)+,R2
    001674  012601                          MOV     (SP)+,R1
407 001676  000207                          RTS     PC
```

```
                                        ;
                                        ;
                                        ;       TEST ROUTINES FOR OUTPUT BUFFER
                                        ;
                                        ;
                                        ;       THIS ROUTINE WRITES AND READS THE TEST PATTERN CONTAINED
                                        ;       IN 'CKDATA'
                                        ;
001700                                  STUFO::
001700  012767  000000  176376          MOV     #0,TD$SW        ;RESET TERM DETECTOR
001706  016667  000002  000000G         MOV     2(SP),PREADD
001714  004767  000420                  1$:     JSR     PC,WRITEO       ;WRITE OUTPUT BUFFER
001720  026667  000004  000000G         CMP     4(SP),PREADD    ;CHECK FOR UPPER LIMITS
001726  103372                          BHIS    1$
                                        ;
001730  016667  000002  000000G         MOV     2(SP),PREADD
001736  004767  000436                  2$:     JSR     PC,READO        ;READ AND COMPARE
001742  026667  000004  000000G         CMP     4(SP),PREADD    ;CHECK FOR UPPER LIMITS
001750  103372                          BHIS    2$
001752  000207                          RTS     PC
                                        ;
                                        ;
                                        ;       TEST 01
                                        ;       WRITE ADDRESS INTO ITSELF
                                        ;
                                        ;
001754                                  T1O::
001754  012767  000000  176376          MOV     #0,TD$SW        ;RESET TERM DETECTOR
001762  016667  000002  000000G         MOV     2(SP),CKDATA    ;ADDRESS IS TEST PATTERN
001770  016667  000002  000000G         MOV     2(SP),PREADD
001776  004767  000336                  1$:     JSR     PC,WRITEO       ;WRITE OUTPUT BUFFER
002002  062767  000002  000000G         ADD     #2,CKDATA       ;BUMP TO NEXT PATTERN (ADDRESS)
002010  026667  000004  000000G         CMP     4(SP),PREADD    ;CHECK FOR UPPER LIMITS
002016  103367                          BHIS    1$
                                        ;
002020  016667  000002  000000G         MOV     2(SP),CKDATA    ;SET TEST PATTERN = ADDRESS
002026  016667  000002  000000G         MOV     2(SP),PREADD
002034  004767  000340                  2$:     JSR     PC,READO        ;READ AND COMPARE
002040  062767  000002  000000G         ADD     #2,CKDATA       ;CHANGE ADDRESS (PATTERN)
002046  026667  000004  000000G         CMP     4(SP),PREADD    ;CHECK FOR UPPER LIMITS
002054  103367                          BHIS    2$
002056  000207                          RTS     PC
                                        ;
                                        ;
                                        ;       TEST 06
                                        ;       MEMORY CROSS-TALK TEST
                                        ;
                                        ;
002060                                  T6O::
                                        ;       MOV     #0,TD$SW        ;RESET TERM DETECTOR
                                        ;       MOV     #10.,R2         ;SET LOOP COUNT
                                        ;       MOV     #-1,CKDATA      ;SET TEST PATTERN TO ALL 1'S
                                        ;10$:   MOV     2(SP),PREADD    ;SET UP START ADDRESS
                                        ;1$:    JSR     PC,WRITEO       ;WRITE OUTPUT BUFFER
                                        ;       ADD     #2,PREADD       ;SKIP AN ADDRESS
                                        ;       CMP     4(SP),PREADD    ;FINISHED
                                        ;       BHIS    1$              ;NO, CONTINUE
```

```
                                        ;       DEC---- R2              ;SUB FROM LOOP COUNT
                                        ;       BNE     10$
                                        ;
                                        ;       CLR     CKDATA          ;SET TEST PATTERN TO ZERO
                                        ;       MOV     2(SP),PREADD    ;RESET START ADDRESS
                                        ;       ADD     #2,PREADD       ;START AT NEXT ADDRESS
                                        ;2$:    JSR     PC,READO        ;READ AND COMPARE
                                        ;       ADD     #2,PREADD       ;BUMP ADDR (SKIP ONE)
                                        ;       CMP     4(SP),PREADD    ;FINISHED
                                        ;       BHIS    2$              ;NO
002060  000207                          ;       RTS     PC
                                        ;
                                        ;
                                        ;       TEST 07
                                        ;       WRITE/READ ALL ADDRESSES WITH ADDRESS COMPLEMENT
                                        ;
                                        ;
002062                                  T70::
002062  012767  000000  176376          ;       MOV     #0,TD$SW        ;RESET TERM DETECTOR
002070  016602  000002                  ;       MOV     2(SP),R2        ;SET R2 = ADDRESS
002074  016667  000002  000000G         ;       MOV     2(SP),PREADD    ;SET UP START ADDRESS
002102  005102                  1$:     ;       COM     R2              ;GET ADDRESS COMPLEMENT
002104  010267  000000G         ;       MOV     R2,CKDATA       ;SET TEST PATTERN = ADDRESS COMPLEMENT
002110  004767  000224          ;       JSR     PC,WRITEO       ;WRITE OUTPUT BUFFER
002114  016702  000000G         ;       MOV     PREADD,R2       ;SET R2 = NEXT ADDRESS
002120  026667  000004  000000G ;       CMP     4(SP),PREADD    ;FINISHED
002126  103365                  ;       BHIS    1$              ;NO, CONTINUE
                                        ;
002130  016602  000002          ;       MOV     2(SP),R2        ;SET R2 = ADDRESS
002134  016667  000002  000000G ;       MOV     2(SP),PREADD    ;RESET START ADDRESS
002142  005102                  2$:     ;       COM     R2              ;GET ADDRESS COMPLEMENT
002144  010267  000000G         ;       MOV     R2,CKDATA       ;SET TEST PATTERN = ADDR COM
002150  004767  000224          ;       JSR     PC,READO        ;READ AND COMPARE
002154  016702  000000G         ;       MOV     PREADD,R2       ;SET R2 = NEXT ADDRESS
002160  026667  000004  000000G ;       CMP     4(SP),PREADD    ;FINICHED
002166  103365                  ;       BHIS    2$              ;NO
002170  000207                  ;       RTS     PC
                                        ;
                                        ;
                                        ;       TEST 12
                                        ;       ADDRESSING TEST
                                        ;
                                        ;
002172                                  T120D::
002172  012767  000000  176376          ;       MOV     #0,TD$SW        ;RESET TERM DETECTOR
002200  016667  000002  000000G         ;       MOV     2(SP),PREADD    ;GET START ADDRESS
002206  016767  000000G 000000G 1$:     ;       MOV     CK2,CKDATA      ;GET READ PATTERN
002214  004767  000160          ;       JSR     PC,READO        ;READ AND COMPARE
002220  162767  000002  000000G ;       SUB     #2,PREADD       ;WRITE AT SAME ADDRESS
002226  016767  000000G 000000G ;       MOV     CK3,CKDATA      ;GET WRITE PATTERN
002234  004767  000100          ;       JSR     PC,WRITEO       ;WRITE OUTPUT BUFFER
002240  026667  000004  000000G ;       CMP     4(SP),PREADD
002246  103357                  ;       BHIS    1$
002250  000207                  ;       RTS     PC
                                        ;
002252                                  T120U::
002252  012767  000000  176376          ;       MOV     #0,TD$SW        ;RESET TERM DETECTOR
```

```
 523 002260  016667  000004  000000G·        MOV·    4(SP),PREADD·    ;START·ADDRESS·=·UPPER·LIMIT
 524 002266  016767  000000G·000000G· 1$:    MOV·    CK2,CKDATA·      ;READ·PATTERN·
 525 002274  004767  000100·                JSR·    PC,READO·        ;READ·AND·COMPARE·
 526 002300  162767  000002  000000G·        SUB·    #2,PREADD·       ;BACK·UP·TO·SAME·ADDRESS·
 527 002306  016767  000000G·000000G·        MOV·    CK3,CKDATA·      ;WRITE·PATTERN·
 528·002314  004767  000020                  JSR·    PC,WRITEO·       ;WRITE·OUTPUT BUFFER·
 529 002320  162767  000004  000000G·        SUB·    #4,PREADD·       ;BACK·UP 1 ADDRESS
 530 002326  026667  000002  000000G·        CMP·    2(SP),PREADD·
 531 002334  003754                          BLE·    1$
 532·002336  000207                          RTS·    PC·
 533                                 ;
 534                                 ;
 535                                 ;
 536                                 ;       WRITE·OUTPUT BUFFER·
 537                                 ;
 538                                 ;
 539 002340  012767  000032  176362· WRITEO: MOV·    #T$OBWA,TD$TAW   ;SET·UP·MEMORY·ADDRESS REG
 540 002346  016767  000000G·176364          MOV·    PREADD,TD$TDW·
 541 002354  012767  000036  176362·         MOV·    #T$OB,TD$TAW·    ;SET·UP·TRANSFER·CODE·
 542·002362·  016767  000000G·176364          MOV·    CKDATA,TD$TDW·   ;WRITE·OUTPUT·BUFFER·A
 543 002370  062767  000002  000000G·        ADD·    #2,PREADD·
 544 002376  000207                          RTS·    PC·
 545                                 ;
 546                                 ;
 547                                 ;       READ·AND·COMPARE·
 548                                 ;
 549                                 ;
 550 002400                          READO:
 551 002400  012767  000034  176362·         MOV·    #T$OBRA,TD$TAW   ;SET·UP·READ ADDRESS·
 552·002406  016767  000000G·176364          MOV·    PREADD,TD$TDW·   ;ADDRESS TO·READ·
 553 002414  012767  000036  176372·         MOV·    #T$OB,TD$TAR·    ;SELECT·OUTPUT·BUFFER·
 554 002422  016701  176374                  MOV·    TD$TDR,R1        ;READ·OUTPUT·BUFFER·
 555 00242C  026701  000000G·                CMP·    CKDATA,R1
 556 002432· 001402·                         BEQ·    .+6              ;OK, CONTINUE·
 557 002434  004767  000010                  JSR·    PC,PRPERO·
 558 002440  062767  000002  000000G·        ADD·    #2,PREADD·       ;ADVANCE ADDRESS·
 559 002446  000207                          RTS·    PC·
 560                                 ;
 561                                 ;
 562·                                ;       PREPARE·ERROR·LIST·
 563                                 ;
 564                                 ;
 565 002450                          PRPERO:
 566 002450                          SAVE·   R1,R2,R3
     002450  010146                  MOV·    R1,-(SP)
     002452  010246                  MOV·    R2,-(SP)
     002454  010346                  MOV·    R3,-(SP)
 567                                 ;
 568 002456  016767  000000G·000000G·        MOV·    PREADD,ERRADD·
 569 002464  012767  000001  000000G·        MOV·    #1,ERRCT         ;PRINT ONE·WORD
 570 002472  012767  000034  176362·         MOV·    #T$OBRA,TD$TAW   ;SET·UP·READ ADDRESS·
 571 002500  016767  000000G·176364          MOV·    PREADD,TD$TDW·   ;ADDRESS TO·READ·
 572·002506  012767  000036  176372·         MOV·    #T$OB,TD$TAR·    ;SELECT·OUTPUT·BUFFER·
 573 002514  016767  176374  000000G·        MOV·    TD$TDR,ERW1      ;READ·OUTPUT·BUFFER·
 574                                 ;
 575                                 ;       RETRY·
 576                                 ;
```

```
577 002522· 012702· 000005                      MOV·    #5,R2·          ;NUMBER·OF·RETRIES
578 002526  012767  000034  176362· 1$:        MOV·    #T$OBRA,TD$TAW  ;SET·UP·READ ADDRESS·
579 002534  016767  000000G·176364              MOV·    PREADD,TD$TDW·  ;ADDRESS TO·READ·
580 002542· 012767  000036  176372·             MOV·    #T$OB,TD$TAR·   ;SELECT·OUTPUT·BUFFER·
581 002550  016703  176374                      MOV·    TD$TDR,R3       ;READ·OUTPUT·BUFFER·
582·002554  020301                              CMP·    R3,R1
583 002556  001003                              BNE·    2$              ;NOT·THE SAME·ERROR·
584 002560  005302·                             DEC·    R2·
585 002562· 001361                              BNE·    1$
586 002564  000403                              BR      3$              ;REPORT·ERROR·
587                                     ;
588 002566  012767  177777  000000G· 2$:       MOV·    #-1,ERRCT       ;INDICATE·UNPREDICTABLE·RESULTS·
589 002574  004767  000000G·          3$:       JSR·    PC,MEMERR·
590                                     ;
591 002600                                      RESTOR· R1,R2,R3
    002600  012603                              MOV·    (SP)+,R3
    002602· 012602·                             MOV·    (SP)+,R2·
    002604  012601                              MOV·    (SP)+,R1
592·002606  000207                              RTS·    PC·
```

```
                                ;
                                ;
                                ;       TEST ROUTINES FOR BYTE TRANSLATOR
                                ;
                                ;
                                ;       THIS ROUTINE WRITES AND READS THE TEST PATTERN CONTAINED
                                ;       IN 'CKDATA'
                                ;
002610                          STUFT::
002610  012767  000000  176376          MOV     #0,TD$SW        ;RESET TERM DETECTOR
002616  016667  000002  000000G         MOV     2(SP),PREADD
002624  004767  000414          1$:     JSR     PC,WRITET       ;WRITE BYTE TRANS
002630  026667  000004  000000G         CMP     4(SP),PREADD    ;CHECK FOR UPPER LIMITS
002636  103372                          BHIS    1$
                                ;
002640  016667  000002  000000G         MOV     2(SP),PREADD
002646  004767  000432          2$:     JSR     PC,READT        ;READ AND COMPARE
002652  026667  000004  000000G         CMP     4(SP),PREADD    ;CHECK FOR UPPER LIMITS
002660  103372                          BHIS    2$
002662  000207                          RTS     PC
                                ;
                                ;
                                ;       TEST 01
                                ;       WRITE ADDRESS INTO ITSELF
                                ;
                                ;
002664                          T1T::
002664  012767  000000  176376          MOV     #0,TD$SW        ;RESET TERM DETECTOR
002672  016667  000002  000000G         MOV     2(SP),CKDATA    ;ADDRESS IS TEST PATTERN
002700  016667  000002  000000G         MOV     2(SP),PREADD
002706  004767  000332          1$:     JSR     PC,WRITET       ;WRITE BYTE TRANS
002712  005267  000000G                 INC     CKDATA
002716  026667  000004  000000G         CMP     4(SP),PREADD    ;CHECK FOR UPPER LIMITS
002724  103370                          BHIS    1$
                                ;
002726  016667  000002  000000G         MOV     2(SP),CKDATA    ;SET TEST PATTERN = ADDRESS
002734  016667  000002  000000G         MOV     2(SP),PREADD
002742  004767  000336          2$:     JSR     PC,READT        ;READ AND COMPARE
002746  005267  000000G                 INC     CKDATA
002752  026667  000004  000000G         CMP     4(SP),PREADD    ;CHECK FOR UPPER LIMITS
002760  103370                          BHIS    2$
002762  000207                          RTS     PC
                                ;
                                ;
                                ;       TEST 06
                                ;       MEMORY CROSS-TALK TEST
                                ;
                                ;
002764                          T6T::
                                ;       MOV     #0,TD$SW        ;RESET TERM DETECTOR
                                ;       MOV     #10.,R2         ;SET LOOP COUNT
                                ;       MOV     #-1,CKDATA      ;SET TEST PATTERN TO ALL 1'S
                                ;10$:   MOV     2(SP),PREADD    ;SET UP START ADDRESS
                                ;1$:    JSR     PC,WRITET       ;WRITE CKDATA TO PREADD
                                ;       INC     PREADD          ;SKIP AND ADDRESS
                                ;       CMP     4(SP),PREADD    ;FINISHED
                                ;       BHIS    1$              ;NO, CONTINUE
```

```
                                        ;       DEC·····  R2·               ;SUB·FROM·LOOP·COUNT·
                                        ;       BNE·      10$
                                        ;
                                        ;       CLR·      CKDATA·           ;SET·TEST·PATTERN·TO·ZERO·
                                        ;       MOV·      2(SP),PREADD·     ;RESET·START·ADDRESS·
                                        ;       INC·      PREADD·           ;START·AT·FIRST·ADDR·+ 1
                                        ;2$:    JSR·      PC,READT·         ;READ·AND·COMPARE·
                                        ;       INC·      PREADD·           ;BUMP·ADDR·(SKIP)
                                        ;       CMP·      4(SP),PREADD·     ;FINISHED·
                                        ;       BHIS·     2$                ;NO·
002764  000207                          ;       RTS·      PC·
                                        ;
                                        ;
                                        ;       TEST·07
                                        ;       WRITE/READ·ALL·ADDRESSES·WITH·ADDRESS·COMPLEMENT·
                                        ;
                                        ;
002766                                  T7T::
002766  012767  000000  176376          ;       MOV·      #0,TD$SW·         ;RESET·TERM·DETECTOR·
002774  016602· 000002                  ;       MOV·      2(SP),R2·         ;SET·R2·=·ADDRESS
003000  016667  000002  000000G·        ;       MOV·      2(SP),PREADD·     ;SET·UP·START·ADDRESS·
003006  005102·                         1$:     COM·      R2·               ;GET·ADDRESS·COMPLEMENT·
003010  010267  000000G·                ;       MOV·      R2,CKDATA·        ;SET·TEST·PATTERN·=·ADDR·COM
003014  004767  000224                  ;       JSR·      PC,WRITET         ;WRITE·'CKDATA'·TO·BYTE·TRANS·
003020  016702· 000000G·                ;       MOV·      PREADD,R2·        ;SET·R2·=·NEXT·ADDRESS
003024  026667  000004  000000G·        ;       CMP·      4(SP),PREADD·     ;FINISHED·
003032  103365                          ;       BHIS·     1$                ;NO, CONTINUE·
                                        ;
003034  016602· 000002                  ;       MOV·      2(SP),R2·         ;SET·R2·=·ADDRESS
003040  016667  000002  000000G·        ;       MOV·      2(SP),PREADD·     ;RESET·START·ADDRESS·
003046  005102·                         2$:     COM·      R2·               ;GET·ADDRESS·COMPLEMENT·
003050  010267  000000G·                ;       MOV·      R2,CKDATA·        ;SET·TEST·PATTERN·=·ADDR·COM
003054  004767  000224                  ;       JSR·      PC,READT·         ;READ·AND·COMPARE·
003060  016702· 000000G·                ;       MOV·      PREADD,R2·        ;SET·R2·=·NEXT ADDRESS
003064  026667  000004  000000G·        ;       CMP·      4(SP),PREADD·     ;FINISHED·
003072  103365                          ;       BHIS·     2$                ;NO·
003074  000207                          ;       RTS·      PC·
                                        ;
                                        ;
                                        ;       TEST·12·
                                        ;       ADDRESSING·TEST·
                                        ;
                                        ;
003076                                  T12TD::
003076  012767  000000  176376          ;       MOV·      #0,TD$SW·         ;RESET·TERM·DETECTOR·
003104  016667  000002  000000G·        ;       MOV·      2(SP),PREADD·     ;GET·START·ADDRESS
003112· 016767  000000G· 000000G·       1$:     MOV·      CK2,CKDATA·       ;GET·READ·PATTERN·
003120  004767  000160                  ;       JSR·      PC,READT·         ;READ·AND·COMPARE·
003124  162767  000001  000000G·        ;       SUB·      #1,PREADD·        ;WRITE·AT·SAME·ADDRESS·
003132· 016767  000000G· 000000G·       ;       MOV·      CK3,CKDATA·       ;GET·WRITE·PATTERN
003140  004767  000100                  ;       JSR·      PC,WRITET
003144· 026667  000004  000000G·        ;       CMP·      4(SP),PREADD·
003152· 103357                          ;       BHIS·     1$
003154  000207                          ;       RTS·      PC·
                                        ;
003156                                  T12TU::
003156  012767  000000  176376          ;       MOV·      #0,TD$SW·         ;RESET·TERM·DETECTOR·
```

```
 708 003164  016667  000004  000000G·          MOV·     4(SP),PREADD·     ;START·ADDRESS·=·UPPER·LIMIT
 709 003172· 016767  000000G·000000G· 1$:      MOV·     CK2,CKDATA·       ;READ·PATTERN·
 710 003200  004767  000100                    JSR·     PC,READT·         ;READ·AND·COMPARE·
 711 003204  162767  000001  000000G·          SUB·     #1,PREADD·        ;BACK·UP TO·SAME·ADDRESS·
 712·003212· 016767  000000G·000000G·          MOV·     CK3,CKDATA·       ;WRITE·PATTERN·
 713 003220  004767  000020                    JSR·     PC,WRITET
 714 003224  162767  000002  000000G·          SUB·     #2,PREADD·        ;BACK·UP 1
 715 003232· 026667  000002· 000000G·          CMP·     2(SP),PREADD·
 716 003240  003754                            BLE·     1$
 717 003242· 000207                            RTS·     PC·
 718                                 ;
 719                                 ;
 720                                 ;
 721                                 ;         WRITE·BYTE·TRANSLATOR·
 722·                                ;
 723                                 ;
 724 003244                          WRITET:
 725 003244  012767  000030  176362·           MOV·     #T$BTAR,TD$TAW    ;SELECT·BYTE·TRANS REG·
 726 003252· 016767  000000G·176364            MOV·     PREADD,TD$TDW·    ;MOVE·ADDR·TO·TRANSFER·REG·
 727 003260  012767  000020  176362·           MOV·     #T$BT,TD$TAW·     ;SELECT·BYTE·TRANS MEMORY·
 728·003266  016767  000000G·176364            MOV·     CKDATA,TD$TDW·    ;WRITE·BYTE·TRANS·
 729 003274  062767  000001  000000G·          ADD·     #1,PREADD·        ;BUMP·ADDRESS·
 730 003302· 000207                            RTS·     PC·
 731                                 ;
 732·                                ;
 733                                 ;         READ·BYTE·TRANSLATOR·
 734                                 ;
 735                                 ;
 736 003304                          READT:
 737 003304  012767  000030  176362·           MOV·     #T$BTAR,TD$TAW    ;SELECT·BYTE·TRANS REG·
 738 003312· 016767  000000G·176364            MOV·     PREADD,TD$TDW·    ;MOVE·ADDR·TO·TRANSFER·REG·
 739 003320  012767  000020  176372·           MOV·     #T$BT,TD$TAR·     ;SELECT·BYTE·TRANS MEMORY·
 740 003326  016701  176374                    MOV·     TD$TDR,R1         ;READ·BYTE·TRANS·
 741 003332· 026701  000000G·                  CMP·     CKDATA,R1         ;CHECK·AGAINST·PATTERN·
 742·003336· 001402·                           BEQ·     .+6               ;OK, CONTINUE·
 743 003340  004767  000010                    JSR·     PC,PRPERT         ;PRINT·ERROR·MESSAGE·
 744 003344  062767  000001  000000G·          ADD·     #1,PREADD·        ;BUMP·ADDRESS·
 745 003352· 000207                            RTS·     PC·
 746                                 ;
 747                                 ;
 748                                 ;         PREPARE·ERROR·LIST·
 749                                 ;
 750                                 ;
 751 003354                          PRPERT:
 752·003354                                    SAVE·    R1,R2,R3
     003354  010146                            MOV·     R1,-(SP)
     003356  010246                            MOV·     R2,-(SP)
     003360  010346                            MOV·     R3,-(SP)
 753                                 ;
 754 003362· 016767  000000G·000000G·          MOV·     PREADD,ERRADD·
 755 003370  012767  000001  000000G·          MOV·     #1,ERRCT·         ;PRINT·ONE·WORD
 756 003376  012767  000030  176362·           MOV·     #T$BTAR,TD$TAW    ;SET·UP·READ·ADDRESS·
 757 003404· 016767  000000G·176364            MOV·     PREADD,TD$TDW·    ;ADDRESS TO·READ·
 758·003412· 012767  000020  176372·           MOV·     #T$BT,TD$TAR·     ;SELE[redacted]
 759 003420  016767  176374  000000G·          MOV·     TD$TDR,ERW1       ;READ·OUTPUT·BUFFER·
 760                                 ;
 761                                 ;         RETRY·
```

```
762·                                    ;
763 003426  012702· 000005                      MOV·    #5,R2·          ;NUMBER·OF·RETRIES
764 003432· 012767  000030· 176362·  1$:        MOV·    #T$BTAR,TD$TAW  ;SET·UP·READ ADDRESS·
765 003440  016767  000000G·176364              MOV·    PREADD,TD$TDW·  ;ADDRESS TO·READ·
766 003446  012767  000020· 176372·             MOV·    #T$BT,TD$TAR·   ;SELECT·BYTE·TRANSLATOR·
767 003454  016703  176374                      MOV·    TD$TDR,R3       ;READ·OUTPUT·BUFFER·
768 003460  020301                              CMP·    R3,R1
769 003462· 001003                              BNE·    2$              ;NOT·THE SAME·ERROR·
770 003464  005302·                             DEC·    R2·
771 003466  001361                              BNE·    1$
772·003470  000403                              BR      3$              ;REPORT·ERROR·
773                                     ;
774 003472· 012767  177777  000000G· 2$:        MOV·    #-1,ERRCT·      ;INDICATE·UNPREDICTABLE·RESULTS·
775 003500  004767  000000G·         3$:        JSR·    PC,MEMERR·
776                                     ;
777 003504                                      RESTOR· R1,R2,R3
    003504  012603                              MOV·    (SP)+,R3
    003506  012602·                             MOV·    (SP)+,R2·
    003510  012601                              MOV·    (SP)+,R1
778 003512· 000207                              RTS·    PC·
779                                     ;
780         000001                              .END·
```

SYMBOL TABLE


```
BITVAL= 000000          DG.SDF= 002000          READX   000636R     005 TD$RST= 176366      T$SEG2= 000001
BIT0  = 000001          DG.TDF= 004000          READ1   000645R     005 TD$SW = 176376      T$SEG3= 000002
BIT1  = 000002          ERRADD= ****** GX       RR1     000742R     005 TD$TAR= 176372      T$SO  = 000001
BIT10 = 002000          ERRCT = ****** GX       SR.ARE  000114      002 TD$TAW= 176362      T$UBUS= 100000
BIT11 = 004000          ERW1  = ****** GX       SR.ARS  000106      002 TD$TDR= 176374      T$1CLK= 000400
BIT12 = 010000          ERW2  = ****** GX       SR.DAY  000010      002 TD$TDW= 176364      T$8BEN= 000020
BIT13 = 020000          ERW3  = ****** GX       SR.DLT  000014      002 T$AD  = 000020      T1F     000072RG    005
BIT14 = 040000          FD.FID  000000      003 SR.ECB  000047      002 T$BA  = 000002      T1I     001140RG    005
BIT15 = 100000          FD.FNB  000006      003 SR.ECH  000046      002 T$BD  = 000010      T1O     001754RG    005
BIT2  = 000004          FD.FVR  000004      003 SR.ECL  000050      002 T$BSO = 100000      T1T     002664RG    005
BIT3  = 000010          FD.LEN  000010      003 SR.FIB  000012      002 T$BT  = 000020      T12FD   000340RG    005
BIT4  = 000020          FN.ACK  000016      004 SR.GRE  000100      002 T$BTAR= 000030      T12FU   000426RG    005
BIT5  = 000040          FN.FSA  000000      004 SR.GRS  000072      002 T$BTD = 002000      T12ID   001366RG    005
BIT6  = 000100          FN.FSB  000002      004 SR.LEN  000122      002 T$CD  = 000100      T12IU   001452RG    005
BIT7  = 000200          FN.FSC  000004      004 SR.LIN  000066      002 T$CLK = 002000      T12OD   002172RG    005
BIT8  = 000400          FN.FSD  000020      004 SR.LIP  000062      002 T$DISK= 000200      T12OU   002252RG    005
BIT9  = 001000          FN.MHR  000010      004 SR.MON  000006      002 T$DRD = 000004      T12TD   003076RG    005
BYTE0 = 000000          FN.NMB  000022      004 SR.NDC  000042      002 T$EMEM= 010000      T12TU   003156RG    005
BYTE1 = 000001          FN.OLS  000006      004 SR.NDS  000036      002 T$FSAA= 000000      T6F     000210RG    005
BYTE2 = 000002          FN.RDC  000014      004 SR.NIN  000030      002 T$FSAB= 000004      T6I     001242RG    005
BYTE3 = 000003          FN.UPD  000012      004 SR.NIP  000022      002 T$FSAC= 000014      T6O     002060RG    005
BYTE4 = 000004          MEMERR= ****** GX       SR.SDB  000032      002 T$FSB2= 000010      T6T     002764RG    005
BYTE5 = 000005          MS.DGN= 010000          SR.SRC  000002      002 T$IB  = 000026      T7F     000212RG    005
BYTE6 = 000006          N     = 000012          SR.SUN  000000      002 T$IBAR= 000024      T7I     001244RG    005
BYTE7 = 000007          N.BFAC= 000004          SR.TWS  000056      002 T$IBE = 020000      T7O     002062RG    005
BYTE8 = 000010          N.BHGH= 000006          SR.WSL  000052      002 T$IBF = 040000      T7T     002766RG    005
BYTE9 = 000011          N.BTCH= 000004          SR.YR   000004      002 T$ICD = 000040      WORD0 = 000000
BYTVAL= 000012          N.BUFB= 004000          SR.1IN  000024      002 T$MODE= 004000      WORD1 = 000002
CALLER  001044R     005 N.BUFW= 002000          SR.1IP  000016      002 T$OB  = 000036      WORD2 = 000004
CF.COT= 000041          N.FOS = 000764          STUFF   000000RG    005 T$OBE = 004000      WORD3 = 000006
CF.DGN= 000046          N.QURY= 000031          STUFI   001052RG    005 T$OBF = 010000      WORD4 = 000010
CF.DHR= 000042          N.SUNT= 000002          STUFO   001700RG    005 T$OBRA= 000034      WORD5 = 000012
CF.DMC= 000047          PREADD= ****** GX       STUFT   002610RG    005 T$OBWA= 000032      WORD6 = 000014
CF.HBR= 000045          PRPERO  002450R     005 TD$CTR= 176370          T$OUTA= 100000      WORD7 = 000016
CF.HRL= 000044          PRPERR  000664R     005 TD$CTW= 176360          T$RBDO= 000200      WORD8 = 000020
CF.UPD= 000043          PRPERT  003354R     005 TD$INL= 004000          T$RNB = 000040      WORD9 = 000022
CKDATA= ****** GX       PRPERI  001602R     005 TD$MEM= 000270          T$RSET= 040000      WRDVAL= 000024
CK2   = ****** GX       READ    000540R     005 TD$OAR= 176344          T$SC  = 000022      WRITE   000514R     005
CK3   = ****** GX       READI   001544R     005 TD$OTR= 176346          T$SCLK= 020000      WRITEO  002340R     005
DBSLEN= 000116          READO   002400R     005 TD$QRD= 000274          T$SEG1= 000000      WRITET  003244R     005
DG.ERR= 001000          READT   003304R     005
```

```
. ABS.  000000     000
        000000     001
SRCOFF  000122     002
FDSCOF  000010     003
FNOFFS  000022     004
TTEST   003514     005
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  2241 WORDS  (9 PAGES)
DYNAMIC MEMORY:  2804 WORDS  ( 10 PAGES)
ELAPSED TIME:  00:00:29
TTEST,TTEST/-SP=[20,1]P,C,TTEST
```

```
TASK    NAME    : ...TMT
PARTITION NAME  : GEN
IDENTIFICATION  : 0736
TASK  UIC       : [7,5]
STACK   LIMITS: 040212 041211 001000 00512.
PRG XFR ADDRESS: 045042.
TOTAL ADDRESS WINDOWS: 2.
TASK  IMAGE  SIZE   : 6240. WORDS
TASK ADDRESS LIMITS: 040000 070253
R-W DISK BLK LIMITS: 000002 000032 000031 00025.

*** ROOT SEGMENT: TMT

R/W MEM  LIMITS: 040000 070253 030254 12460.
DISK BLK LIMITS: 000002 000032 000031 00025.

MEMORY ALLOCATION SYNOPSIS:

SECTION                                          TITLE   IDENT   FILE
-------                                          -----   -----   ----
. BLK.  :(RW,I,LCL,REL,CON) 041212 016562 07538.
                            041212 011506 04934. TMT             TMT.OBJ;1
FDSCOF:(RW,I,LCL,ABS,CON)   000000 000000 00000.
                            000000 000000 00000. TMT             TMT.OBJ;1
                            000000 000000 00000. TTEST           TTEST.OBJ;1
FNOFFS:(RW,I,LCL,ABS,CON)   000000 000000 00000.
                            000000 000000 00000. TMT             TMT.OBJ;1
                            000000 000000 00000. TTEST           TTEST.OBJ;1
SRCOFF:(RW,I,LCL,ABS,CON)   000000 000000 00000.
                            000000 000000 00000. TMT             TMT.OBJ;1
                            000000 000000 00000. TTEST           TTEST.OBJ;1
TTEST. :(RW,I,LCL,REL,CON)  057774 003514 01868.
                            057774 003514 01868. TTEST           TTEST.OBJ;1
$$FSR1:(RW,D,GBL,REL,OVR)   063510 001020 00528.
                            063510 001020 00528. TMT             TMT.OBJ;1
$$FSR2:(RW,D,GBL,REL,CON)   064530 000104 00068.
$$RESL:(RW,I,LCL,REL,CON)   064634 003420 01808.
$$RESM:(RW,I,LCL,REL,CON)   132000 007656 04014.


GLOBAL SYMBOLS:

CKDATA 042506-R  ERW1   042522-R  STUFF  057774-R  TiI    061134-R  Ti2ID  061362-R  Ti2TU  063152-R  T7F    060206-R
CK2    042510-R  ERW2   042524-R  STUFI  061046-R  TiO    061750-R  Ti2IU  061446-R  T6F    060204-R  T7I    061240-R
CK3    042512-R  ERW3   042526-R  STUFO  061674-R  TiT    062660-R  Ti2OD  062166-R  T6I    061236-R  T7O    062056-R
ERRADD 042516-R  MEMERR 047752-R  STUFT  062604-R  Ti2FD  060334-R  Ti2OU  062246-R  T6O    062054-R  T7T    062762-R
ERRCT  042520-R  PREADD 042514-R  TiF    060066-R  Ti2FU  060422-R  Ti2TD  063072-R  T6T    062760-R
```

*** TASK BUILDER STATISTICS:

TOTAL WORK FILE REFERENCES: 11893.
WORK FILE READS: 0.
WORK FILE WRITES: 0.
SIZE OF CORE POOL: 6634. WORDS (25. PAGES)
SIZE OF WORK FILE: 2048. WORDS (8. PAGES)

ELAPSED TIME:00:00:15

```
                        .TITLE  TMAIN
                        .LIST   MEB


                ;
                ;
                ;       TERM DETECTOR 'MANUAL' DEBUGGING AIDS.
                ;       MAIN MODULE.
                ;
                ;
                ;       THIS MODULE PASSES CONTROL TO ITS SUB-MODULES (COMMAND
                ;       MODULES) BASED ON INFORMATION IN THE COMMAND LINE (TERMINAL
                ;       INPUT). THE SUB-MODULES CONTINUE TO PARSE THE COMMAND LINE,
                ;       TRANSFERRING CONTROL TO THEIR SUB-ROUTINES. THE SUB-ROUTINES
                ;       DO THE ACTUAL INTERFACING WITH THE HARDWARE. THUS, THERE ARE
                ;       THREE LEVELS OF CONTROL. FOR EXAMPLE, TAKE THE COMMAND:
                ;       >LD OB 0
                ;       THE 'LD' REPRESENTS THE FIRST LEVEL OF CONTROL AND IS PARSED
                ;       BY THE MODULE TMAIN. THE 'OB' REPRESENTS THE SECOND LEVEL OF
                ;       CONTROL AND IS PARSED BY THE TMAIN SUB-MODULE TMEM. THE THIRD
                ;       LEVEL OF CONTROL, A SUB-ROUTINE OF TLOAD ACTUALLY CONTROLS
                ;       THE LOADING OF THE HARDWARE. TMAIN CONTAINS THE COMMANDS
                ;       'RS' (RESET) AND 'EX' (EXIT).
                ;
                ;
                ;       SUB-MODULES OF TMAIN:
                ;       TREG    REGISTER COMMANDS
                ;       TMEM    MEMORY COMMANDS
                ;       TDATA   INPUT BUFFER AND BYTE TRANSLATOR COMMANDS
                ;       TRUN    RUN TD COMMANDS

                ;       CONTROL IS RETURNED TO TMAIN WHEN A COMMAND HAS BEEN
                ;       EXECUTED OR ANND ERROR ENCOUNTERED.

                ;       TMAIN ALSO CONTAINS DATA AND SUBROUTINES COMMON TO ALL OF ITS
                ;       SUB-MODULES.


                ;       ASSEMBLY:
                ;       MCR>MAC TMAIN,LP=IM04,TMAIN             FROM [5,3]

                ;       TASK BUILD:
                ;       1. TD STAND-ALONE PACK COMMAND FILE AIDTD.CMD
                ;       AIDTD/DA,AIDTD=TMAIN,TREG,TDATA,TRUN
                ;       /
                ;       PAR=PAR14K
                ;       ASG=TT0:1:2
                ;       /
                ;       //

                ;       2. NPIC SYSTEM COMMAND FILE AIDTD.CMD (USING OVERLAYS)
                ;          ************** NOTE **************
                ;          TASK BUILD MUST BE DONE ON NPIC /04
                ;          MAPPING DOES NOT COME OUT RIGHT WHEN TKB IS DONE ON V4.5
                ;          EG. ADDRESS OF SYSTEM ROUTINE $DIV COMES OUT INCORRECTLY
                ;          ************** NOTE **************
                ;
                ;       AIDTD,AIDTD=AIDTD/MP
                ;       PAR=GEN:40000:40000
```

```
58                              ;       RESLIB=[1,1]FCSRES/RO
59                              ;       ASG=TT0:1:2
60                              ;       //
61                              ;
62                              ;       OVERLAY DESCRIPTION AIDTD:ODL:
63                              ;       .ROOT   TMAIN-*(TREG,TMEM,TDATA,TRUN)
64                              ;       .END
```

```
                                ;
                                ;
                                ;       LOCAL DATA AREAS
                                ;
                                ;
                                        .MCALL  QIOW$S,QIO$S,EXIT$S,ABRT$S,GCML$,GCMLB$,FSRSZ$,CLEF$S
                                        .MCALL  ASTX$S,FDBDF$,FDRC$A,FDOP$A,NMBLK$,OPEN$R,GET$,CLOSE$
                                ;
                                        .GLOBL  IO.WVB,IO.RVB,IO.ATA,IO.DET
                                ;
                                ;
000001                          EFN.1   ==      1               ;EVENT FLAG FOR TERMINAL QIO'S
000001                          LUN.TT  ==      1               ;TERMINAL LUN
000002                          CMILUN  ==      2               ;GCML LUN
000003                          TXTLUN  ==      3               ;LUN FOR DISK 'DATA' FILE
000004                          TRLUN   ==      4               ;LUN FOR DISK BYTE TRANSLATOR OVERRIDE FILE
                                ;
000004                          LOOP    ==      4               ;LOOP ON TEST FLAG
000010                          ONCE    ==      10              ;PRINT ONE WORD ONLY
000020                          ASTFLG  ==      20              ;QIO +AST ISSUED
000100                          RIDE    ==      100             ;BYTE TRANSLATOR OVERRIDE IN EFFECT
000200                          RP      ==      200             ;REPEAT PROMPT
000400                          QP      ==      400             ;PRINT OUTPUT BUFFER CONTENTS
001000                          QN      ==      1000            ;DO NOT PRINT OB CONTENTS
002000                          OUT     ==      2000            ;CONTROL PRINTING OF MEMORY CONTENTS
                                ;
                                ;
                                        .NLIST  BEX
000000  003654  076640          MYSELF: .RAD50  /AIDTD/
000004                          STAT::  .BLKW   2
000010  000000                  ERWORD: .WORD   0               ;INDEX VALUE FOR ERROR MESSAGE TABLE
000012  000000                  BINWD:: .WORD   0               ;TARGET FOR NUMERIC CONVERSIONS FROM ASCII
000014  000000                  BASE::  .WORD   0               ;ALL PURPOSE FLAG
000016  000000                  APLACE:: .WORD  0               ;PRELIM BIT SETTINGS FOR TD CONTROL REG (TD$CTW)
000020                          GCMBUF:: .BLKW  41.             ;COMMAND LINE BUFFER
000142  000000                  GCMLEN:: .WORD  0               ;COMMAND LINE LENGTH
000144  000000                  GCMPNT:: .WORD  0               ;COMMAND LINE POINTER
000146  000000                  ASTWRD: .WORD   0               ;RECEIVER FOR AST CHAR
000150  000000                  RTNPT:: .WORD   0               ;RTN ADDR SAVE AREA
000152  000000                  MSTRT:: .WORD   0               ;START ADDR FOR MEMORY LOADING/PRINTING
000154  000000                  MSTR2:: .WORD   0               ;WORKING ADDR FOR LOADING/PRINTING
000156  000000                  MEND::  .WORD   0               ;END ADDRESS FOR MEMORY LOADING/PRINTING
000160  000000                  INCVAL:: .WORD  0               ;MEMORY INCREMENT VALUE
000162  000000                  RSPCNT:: .WORD  0               ;(COMMAND LINE) RESPONSE COUNT
000164  000000                  UPLIM:: .WORD   0               ;MEMORY UPPER LIMIT
000166  000000                  COUNT:: .WORD   0               ;COUNTER FOR 8 TO 6-BIT CONVERSION
000170  000000                  DATALN:: .WORD  0               ;LENGTH OF 'DATA' BUFFER
000172  000000                  DPLUS:: .WORD   0               ;WORK COUNTER FOR 'DATA' RTN
000174  000000                  OFF6::  .WORD   0               ;OFFSET INTO 'DATA' BUFFER
000176  000000                  CHLEN:: .WORD   0               ;LENGTH OF CHANGE DATA + CHANGE START ADDRESS
                                ;
000200                          WWORDS::                        ;CONVERTED NUMERIC VALUES (FROM COMM LINE)
000200  000000                  DATA1:: .WORD   0
000202  000000                  DATA2:: .WORD   0
000204  000000                  DATA3:: .WORD   0
                                ;
                                ;       MEMORY LIMITS
```

```
                                        ;
000206  007777                AHIGH:: .WORD   7777            ;FSA A
000210  000000                ALOW::  .WORD   0
000212  007777                BHIGH:: .WORD   7777            ;FSA B
000214  000000                BLOW::  .WORD   0
000216  007777                CHIGH:: .WORD   7777            ;FSA C
000220  000000                CLOW::  .WORD   0
000222  163776                IHIGH:: .WORD   163776          ;INPUT BUFFER
000224  160000                ILOW::  .WORD   160000
000226  003776                OHIGH:: .WORD   3776            ;OUTPUT BUFFER
000230  000000                OLOW::  .WORD   0
000232  000377                THIGH:: .WORD   255.            ;BYTE TRANSLATOR
000234  000000                TLOW::  .WORD   0
                                        ;
                                        ;
                                        ;       ROUTINE CONTROL TABLES
                                        ;       COMMAND MNEMONICS AND ASSOCIATED ROUTINE ADDRESSES
                                        ;
                                        ;
000236                          CTBL::
000236      123     124                 .ASCII  /ST/            ;STORE INTO A REGISTER
000240  000000G                         .WORD   ST1
000242      122     105                 .ASCII  /RE/            ;READ A REGISTER
000244  000000G                         .WORD   RE1
000246      114     104                 .ASCII  /LD/            ;LOAD A MEMORY
000250  000000G                         .WORD   LD1
000252      120     122                 .ASCII  /PR/            ;PRINT MEMORY CONTENTS TO CONSOLE
000254  000000G                         .WORD   PR1
000256      104     101                 .ASCII  /DA/            ;SIMULATE DATA BASE
000260  000000G                         .WORD   DA1
000262      103     110                 .ASCII  /CH/            ;CHANGE SIMULATED DATA
000264  000000G                         .WORD   CH1
000266      114     111                 .ASCII  /LI/            ;LIST SIMULATED DATA
000270  000000G                         .WORD   LI1
000272      105     116                 .ASCII  /EN/            ;CONTROL LENGTH OF SIMULATED DATA
000274  000000G                         .WORD   EN1
000276      104     106                 .ASCII  /DF/            ;READ SIMULATED DATA OR BYTE TRANSLATOR
000300  000000G                         .WORD   DF1             ;DATA IN FROM DISK
000302      124     122                 .ASCII  /TR/            ;TRANSFER (BYTE TRANS OR INPUT BUFFER)
000304  000000G                         .WORD   TR1             ;FROM PROGRAM MEMORY TO TD
000306      122     123                 .ASCII  /RS/            ;RESET TERM DETECTOR
000310  010606'                         .WORD   RS1
000312      122     125                 .ASCII  /RU/            ;RUN TERM DETECTOR
000314  000000G                         .WORD   RU1
000316      105     130                 .ASCII  /EX/            ;EXIT TD EXERCISER
000320  010620'                         .WORD   EX1
000015                          CNUM    ==      <.-CTBL>/4
```

```
                                        ;
                                        ;
                                        ;       'DATA' TABLES
                                        ;       DTBL    =       6-BIT TABLE
                                        ;       DSAVE   =       ASCII MIRROR OF 8-BIT INPUT
                                        ;
                                        ;
000322                          DTBL::  .BLKW   <<1364.*3>/8.>+1
002322                          DSAVE:: .BLKB   1364.
005046                          DSEND::
005046  000                             .BYTE   0
                                        .EVEN
                                        ;
                                        ;       CONVERSION TABLE FOR 'DATA'
                                        ;
005050                          TR6TBL::
005050  377     377     377             .BYTE   -1,-1,-1,-1,-1          ;CODES 0 - 11 UNUSED
005055  377     377     377             .BYTE   -1,-1,-1,-1,-1
005062  075                             .BYTE   75                      ;LINE FEED
005063  377     377                     .BYTE   -1,-1                   ;CODES 13 -14 UNUSED
005065  074                             .BYTE   74                      ;CARRAIGE RETURN
005066  377     377     377             .BYTE   -1,-1,-1,-1,-1,-1,-1    ;CODES 16 -37 UNUSED
005074  377     377     377             .BYTE   -1,-1,-1,-1,-1,-1,-1
005102  377     377     377             .BYTE   -1,-1,-1,-1,-1,-1,-1
005110  040                             .BYTE   40                      ;SPACE
005111  041                             .BYTE   41                      ;EXCLAMATION POINT
005112  042                             .BYTE   42                      ;QUOTES
005113  043                             .BYTE   43                      ;POUND SIGN
005114  044                             .BYTE   44                      ;DOLLAR
005115  045                             .BYTE   45                      ;PERCENT
005116  046                             .BYTE   46                      ;AMPERSAND
005117  047                             .BYTE   47                      ;APOSTROPHE
005120  050                             .BYTE   50                      ;LEFT PAREN
005121  051                             .BYTE   51                      ;RIGHT PAREN
005122  052                             .BYTE   52                      ;ASTERISK
005123  053                             .BYTE   53                      ;PLUS
005124  054                             .BYTE   54                      ;COMMA
005125  055                             .BYTE   55                      ;HYPHEN
005126  056                             .BYTE   56                      ;PERIOD
005127  057                             .BYTE   57                      ;SLASH
005130  060                             .BYTE   60                      ;0
005131  061                             .BYTE   61                      ;1
005132  062                             .BYTE   62                      ;2
005133  063                             .BYTE   63                      ;3
005134  064                             .BYTE   64                      ;4
005135  065                             .BYTE   65                      ;5
005136  066                             .BYTE   66                      ;6
005137  067                             .BYTE   67                      ;7
005140  070                             .BYTE   70                      ;8
005141  071                             .BYTE   71                      ;9
005142  072                             .BYTE   72                      ;COLON
005143  073                             .BYTE   73                      ;SEMI-COLON
005144  377                             .BYTE   -1                      ;LEFT ANGLE BRACKET NOT USED
005145  377                             .BYTE   -1                      ;EQUAL SIGN NOT USED
005146  377                             .BYTE   -1                      ;RIGHT ANGLE BRACKET NOT USED
005147  077                             .BYTE   77                      ;QUESTION MARK
005150  000                             .BYTE   0                       ;USE AT SIGN FOR DOCUMENT START
```

```
005151   001          .BYTE  01      ;A
005152   002          .BYTE  02      ;B
005153   003          .BYTE  03      ;C
005154   004          .BYTE  04      ;D
005155   005          .BYTE  05      ;E
005156   006          .BYTE  06      ;F
005157   007          .BYTE  07      ;G
005160   010          .BYTE  10      ;H
005161   011          .BYTE  11      ;I
005162   012          .BYTE  12      ;J
005163   013          .BYTE  13      ;K
005164   014          .BYTE  14      ;L
005165   015          .BYTE  15      ;M
005166   016          .BYTE  16      ;N
005167   017          .BYTE  17      ;O
005170   020          .BYTE  20      ;P
005171   021          .BYTE  21      ;Q
005172   022          .BYTE  22      ;R
005173   023          .BYTE  23      ;S
005174   024          .BYTE  24      ;T
005175   025          .BYTE  25      ;U
005176   026          .BYTE  26      ;V
005177   027          .BYTE  27      ;W
005200   030          .BYTE  30      ;X
005201   031          .BYTE  31      ;Y
005202   032          .BYTE  32      ;Z
005203   033          .BYTE  33      ;LEFT SQ BRACKET - ZONE
005204   034          .BYTE  34      ;BACK SLASH - SUBZONE
005205   035          .BYTE  35      ;RGT SQ BRACKET - PARA
005206   036          .BYTE  36      ;UP ARROW - SENT
005207   037          .BYTE  37      ;UNDERBAR - CLOSE UP
                                     ;CODES 140 - 177 NOT USED
                      .NLIST MEB
         000040       .REPT  40
                      .BYTE  -1
                      .ENDR
                      .LIST  MEB
                      .EVEN
```

```
                                ;
                                ;
                                ;       DEFAULT BYTE TRANSLATOR TABLE
                                ;
                                ;
005250                  BTRANS::
005250  002077                  .WORD   002077          ;DOCUMENT START
005252  000001                  .WORD   001             ;A              1
005254  000002                  .WORD   002             ;B              2
005256  000003                  .WORD   003             ;C              3
005260  000004                  .WORD   004             ;D              4
005262  000005                  .WORD   005             ;E              5
005264  000006                  .WORD   006             ;F              6
005266  000007                  .WORD   007             ;G              7
005270  000010                  .WORD   010             ;H              10
005272  000011                  .WORD   011             ;I              11
005274  000012                  .WORD   012             ;J              12
005276  000013                  .WORD   013             ;K              13
005300  000014                  .WORD   014             ;L              14
005302  000015                  .WORD   015             ;M              15
005304  000016                  .WORD   016             ;N              16
005306  000033                  .WORD   033             ;O              17
005310  000034                  .WORD   034             ;P              20
005312  000021                  .WORD   021             ;Q              21
005314  000022                  .WORD   022             ;R              22
005316  000023                  .WORD   023             ;S              23
005320  000024                  .WORD   024             ;T              24
005322  000025                  .WORD   025             ;U              25
005324  000026                  .WORD   026             ;V              26
005326  000027                  .WORD   027             ;W              27
005330  000030                  .WORD   030             ;X              30
005332  000031                  .WORD   031             ;Y              31
005334  000032                  .WORD   032             ;Z              32
005336  040033                  .WORD   040033          ;ZONE MARK      33
005340  100034                  .WORD   100034          ;SUBZONE MARK   34
005342  010035                  .WORD   010035          ;PARAGRAPH MARK 35
005344  004036                  .WORD   004036          ;SENTENCE       36
005346  000437                  .WORD   437             ;CLOSE UP       37
005350  000400                  .WORD   BIT8            ;SPACE          40
005352  000400                  .WORD   BIT8            ;EXCL POINT     41
005354  000400                  .WORD   BIT8            ;QUOTE          42
005356  000301                  .WORD   301             ;POUND          43
005360  000302                  .WORD   302             ;DOLLAR         44
005362  000400                  .WORD   BIT8            ;PERCENT        45
005364  000400                  .WORD   BIT8            ;AMPERSAND      46
005366  000400                  .WORD   BIT8            ;APOSTROPHE     47
005370  000400                  .WORD   BIT8            ;LEFT PAREN     50
005372  000400                  .WORD   BIT8            ;RIGHT PAREN    51
005374  000400                  .WORD   BIT8            ;ASTERISK       52
005376  000400                  .WORD   BIT8            ;PLUS           53
005400  000400                  .WORD   BIT8            ;COMMA          54
005402  000400                  .WORD   BIT8            ;HYPHEN         55
005404  000400                  .WORD   BIT8            ;PERIOD         56
005406  000400                  .WORD   BIT8            ;SLASH          57
005410  000257                  .WORD   257             ;0              60
005412  000277                  .WORD   277             ;1              61
005414  000057                  .WORD   057             ;2              62
```

```
 324 005416  000077                    .WORD   077             ;3              63
 325 005420  000117                    .WORD   117             ;4              64
 326 005422  000137                    .WORD   137             ;5              65
 327 005424  000157                    .WORD   157             ;6              66
 328 005426  000177                    .WORD   177             ;7              67
 329 005430  000217                    .WORD   217             ;8              70
 330 005432  000237                    .WORD   237             ;9              71
 331 005434  000400                    .WORD   BIT8            ;COLON          72
 332 005436  000400                    .WORD   BIT8            ;SEMI-COLON     73
 333 005440  000400                    .WORD   BIT8            ;               74
 334 005442  000400                    .WORD   BIT8            ;               75
 335 005444  000303                    .WORD   303             ;               76
 336 005446  000400                    .WORD   BIT8            ;DECIMAL        77
 337 005450      000             BTEND:: .BYTE   0
 338                                     .EVEN
 339                             ;
 340                             ;       BYTE TRANSLATOR OVERRIDE TABLE
 341                             ;
 342 005452                      BTOVER::
 343 005452                              .BLKW   <BTEND-BTRANS>/2
```

```
                                        ;
                                        ;
                                        ;       PRINT LINE.
                                        ;
005652  015     012                     .BYTE   15,12                   ;PRECEDE PRINT LINE WITH CRLF.
005654                          PRINT::
                                        .N_IST  MEB
        000116                          .REPT   78.
                                        .BYTE   40
                                        .ENDR
                                        .LIST   MEB
                                        ;
                                        ;
                                        ;       TABLE OF MESSAGES
                                        ;
                                        ;
005772  000                             .BYTE   0
005773  015     012     015             .BYTE   15,12,15,12
005777  124     105     123             .ASCIZ  /TEST(S) ENDED/
006015  015     012                     .BYTE   15,12
006017  105     116     104             .ASCIZ  /END OF FILE REACHED/
006043  015     012                     .BYTE   15,12
006045  105     116     104             .ASCIZ  /END OF MEMORY REACHED/
006073  015     012                     .BYTE   15,12
006075  105     116     124             .ASCIZ  /ENTER ANY CHARACTER TO EXIT LOOP/-
006136  015     012     015             .BYTE   15,12,15,12,15,12
006144  124     105     122             .ASCII  /TERM DETECTOR MANUAL DEBUGGING AIDS/
006207  015     012     000             .BYTE   15,12,0
006212  015     012                     .BYTE   15,12
006214  111     114     114             .ASCIZ  /ILLEGAL ODD ADDRESS/
006240  015     012                     .BYTE   15,12
006242  106     117     122             .ASCIZ  /FOR BT, 'CHANGE' COMMAND MUST PRECEED 'LIST'/
006317  015     012                     .BYTE   15,12
006321  111     114     114             .ASCIZ  /ILLEGAL CHARACTER IN 'DATA'/
006355  015     012                     .BYTE   15,12
006357  123     124     101             .ASCIZ  /START ADDRESS OUT OF RANGE HIGH/
006417  015     012                     .BYTE   15,12
006421  123     124     101             .ASCIZ  /START ADDRESS OUT OF RANGE LOW/
006460  015     012                     .BYTE   15,12
006462  111     116     126             .ASCIZ  /INVALID TABLE NAME/
006505  015     012                     .BYTE   15,12
006507  102     131     124             .ASCIZ  /BYTE TRANSLATOR TABLE OVERFLOW/
006546  015     012                     .BYTE   15,12
006550  105     115     120             .ASCIZ  /EMPTY INPUT LINE IN BYTE TRANSLATOR FILE/
006621  015     012                     .BYTE   15,12
006623  105     122     122             .ASCIZ  /ERROR ON READ OF BYTE TRANSLATOR FILE/
006671  015     012                     .BYTE   15,12
006673  105     122     122             .ASCIZ  /ERROR ON OPEN OF BYTE TRANSLATOR FILE/
006741  015     012                     .BYTE   15,12
006743  105     122     122             .ASCIZ  /ERROR ON READ OF 'DATA' FILE/
007000  015     012                     .BYTE   15,12
007002  105     122     122             .ASCIZ  /ERROR ON OPEN OF 'DATA' FILE/
007037  015     012                     .BYTE   15,12
007041  111     116     126             .ASCIZ  /INVALID FILE MNEMONIC/
007067  015     012                     .BYTE   15,12
007071  047     104     101             .ASCIZ  /'DATA' COMMAND MUST PRECEED 'END'/
007133  015     012                     .BYTE   15,12
```

```
402 007135     047     114     111          .ASCIZ  /'LIST' START ADDRESS OUT OF RANGE/
403 007177     015     012                  .BYTE   15,12
404 007201     047     104     101          .ASCIZ  /'DATA' COMMAND MUST PRECEED 'LIST'/
405 007244     015     012                  .BYTE   15,12
406 007246     047     104     101          .ASCIZ  /'DATA' COMMAND MUST PRECEED 'CHANGE'/
407 007313     015     012                  .BYTE   15,12
408 007315     111     116     126          .ASCIZ  /INVALID MEMORY MNEMONIC/
409 007345     015     012                  .BYTE   15,12
410 007347     111     116     126          .ASCIZ  /INVALID REGISTER MNEMONIC/
411 007401     015     012                  .BYTE   15,12
412 007403     111     116     126          .ASCIZ  /INVALID NUMERIC VALUE/
413 007431     015     012                  .BYTE   15,12
414 007433     115     111     123          .ASCIZ  /MISSING OPERAND/
415 007453     015     012                  .BYTE   15,12
416 007455     111     116     126          .ASCIZ  /INVALID COMMAND/
417 007475     015     012                  .BYTE   15,12
418 007477     111     116     126          .ASCIZ  /INVALID LOOP CONTROL/
419 007524     015     012                  .BYTE   15,12
420 007526     105     116     124          .ASCIZ  /ENTER CORRECT RESPONSE/
421 007555     015     012                  .BYTE   15,12
422 007557     105     116     124          .ASCIZ  /ENTER COMMAND/
423 007575     015     012                  .BYTE   15,12
424 007577     123     105     114          .ASCIZ  /SELECT BASE/
425 007613     377                  ASCIZ:  .BYTE   377
426                                         .EVEN
427                                         .LIST   BEX
428                                         .NLIST  CND
429                                 ;
430                                 ;
431                                 ;       COMMAND LINE MACRO
432                                 ;
433                                 ;
434 007614                          GCMBLK: GCMLB$  2,,GCMBUF,CMILUN
    007614     002                          .BYTE   2
    007656     002                          .BYTE   CMILUN
    007662 010002'                          .WORD   65$
    007670     001                          .BYTE   1
    007633     000C                         .BYTE   FD.TTY!FD.REC!FD.CCL
    007754     000                          .BYTE   0
    007755     047                          .BYTE   GE.COM!GE.IND!GE.CLO!GE.SIZ
    007766     377     002                  .BYTE   -1,2
    007770 010042'                          .WORD   64$
    007772 000020'                          .WORD   GCMBUF
    007774     015     012                  .ASCII  <15><12>
    007776     040     040     040          .ASCII  /   /
    010001     076
    010010  012321                          .RAD50  /CMI/
    010012  000000                          .WORD   0
    010014  000000                          .WORD   0
    010016  012314                          .RAD50  /CMD/
    010040  000120                          .WORD   80.
435                                 ;
436                                 ;       FDB FOR DISK 'DATA' FILE
437                                 ;
438 010122                          TXTFDB:: FDBDF$
439 010262                                  FDRC$A  ,GCMBUF,80.
    010144 000020'                          .WORD   GCMBUF
```

```
        010142  000120                          .WORD   80.
    440 010262                                  FDOP$A  TXTLUN,,TXTNMB
        010164      003                         .BYTE   TXTLUN
        010170  010262'                         .WORD   TXTNMB
    441 010262                          TXTNMB: NMBLK$  DATA.TXT
        010270  014474  003100                    .RAD50  /DATA/
        010274  000000                          .WORD   0
        010276  100324                          .RAD50  /TXT/
    442
    443                                 ;
    444                                 ;       FDB FOR BYTE TRANSLATOR OVERRIDE
    445 010320                          ;
    446 010460                          TRFDB:: FDBDF$
        010342  000020'                         FDRC$A  ,GCMBUF,80.
        010340  000120                          .WORD   GCMBUF
    447 010460                                  .WORD   80.
        010362      004                         FDOP$A  TRLUN,,TRNMB
        010366  010460'                         .BYTE   TRLUN
    448 010460                          TRNMB:  .WORD   TRNMB
        010466  077721  055170                  NMBLK$  TRANS,TXT
        010472  000000                          .RAD50  /TRANS/
        010474  100324                          .WORD   0
    449 010516                                  .RAD50  /TXT/
                                                FSRSZ$  2
```

```
                                        ;
                                        ;
                                        ;       ENTER HERE
                                        ;
                                        ;
010516                                  START:
010516  004767  003052                          JSR     PC,OUT1         ;ISSUE INFORMATION MESSAGE
                                        ;
                                        ;
                                        ;
                                        ;       TOP OF COMMAND LOOP
                                        ;
                                        ;
010522                                  COMXX::
010522  004767  003212                          JSR     PC,SELTST       ;PROMPT FOR COMMAND
010526  004767  002346                          JSR     PC,FIND         ;FIND THE COMMAND
010532  103003                                  BCC     1$              ;OK, CONTINUE
010534  004767  003164                          JSR     PC,ERR2         ;'INVALID COMMAND'
010540  000770                                  BR      COMXX
010542  022700  000002                  1$:     CMP     #2,R0           ;COMMANDS ARE 2 CHARS
010546  001403                                  BEQ     2$
010550  004767  003150                          JSR     PC,ERR2         ;'INVALID COMMAND'
010554  000762                                  BR      COMXX           ;TRY AGAIN
                                        ;
010556  012700  000015                  2$:     MOV     #CNUM,R0        ;R0 = NUMBER OF COMMANDS
010562  012702  000236                          MOV     #CTBL,R2        ;R2 -> TABLE OF VALID COMMANDS
010566  004767  002214                          JSR     PC,SCAN         ;FIND MATCH IN TABLE
010572  103003                                  BCC     3$              ;OK, CONTINUE
010574  004767  003124                          JSR     PC,ERR2         ;COMMAND NOT IN TABLE
010600  000750                                  BR      COMXX           ;TRY AGAIN
                                        ;
                                        ;       JUMP TO THE ROUTINE THAT GOVERNS THE COMMAND
                                        ;
010602  000171  000000                  3$:     JMP     @(R1)
```

```
                                        ;
                                        ;
                                        ;       RESET
                                        ;
                                        ;
010606                                  RS1:
010606  012767  000000  176376                  MOV     #0,TD$SW        ;RESET TD
010614  000167  177702                          JMP     COMXX
                                        ;
                                        ;
                                        ;
                                        ;       EXIT
                                        ;
                                        ;
010620                                  EX1:
010620                                          EXIT$S
010620  012746                                  MOV     (PC)+,-(SP)
010622      063     001                         .BYTE   51.,1
010624  104377                                  EMT     ^O<377>
```

```
                                ;
                                ;
                                ;       SET UP BEFORE PROMPTING ON ANY 'LD' COMMAND OR BEFORE
                                ;       EXECUTING ANY 'PR' COMMAND FOR FSA A, B, C. THE FIELDS
                                ;       SET UP HERE ARE USED IN THE TMAIN PROMPTING SUBROUTINE
                                ;       PDATA (LDCNTL FOR FSA'S) AND THE PRINT ROUTINE PRDATA
                                ;       (PRCNTL FOR FSA'S).
                                ;
                                ;       INPUT:
                                ;       2(SP)   LOAD/PRINT START ADDRESS
                                ;       4(SP)   LOWER MEMORY LIMIT
                                ;       6(SP)   UPPER MEMORY LIMIT
                                ;       APLACE  - FSA SELECTION
                                ;
                                ;       OUTPUT:
                                ;       INCVAL  - VALUE BY WHICH MEMORY ADDRESS INCREMENTS
                                ;       MEND    - ADDRESS OF LAST WORD IN MEMORY TO BE LOADED/PRINTED
                                ;       RSPCNT  - NUMBER OF WORDS TO EXPECT IN COMMAND LINE (MEMORY WIDTH)
                                ;
                                ;       C-BIT CLEAR     NO ERROR (START ADDRESS IS IN RANGE)
                                ;       C-BIT SET       ERROR IN RANGE
                                ;
                                ;
                                ;       MEMORY          INCVAL          RSPCNT
                                ;       FSA A           1               3
                                ;       FSA B           1               3
                                ;       FSA C           1               3
                                ;
                                ;
010626                          BUFFSA::
010626  012767  000000  176376          MOV     #0,TD$SW        ;RESET TD
010634  052767  020000  167154          BIS     #T$SCLK,APLACE  ;START CLOCK
010642  016767  167150  176360          MOV     APLACE,TD$CTW   ;SET TD CONTROL REG
                                ;
                                ;       IF THE FIELD MEND IS NOT =-1, THEN A 'LD' OR
                                ;       'PR' ROUTINE HAS PUT A MEMORY UPPER LIMIT IN
                                ;       THIS FIELD, EG, A COMMAND SUCH AS:
                                ;       >LD AM 0 10
                                ;       WOULD PUT 10 IN THE FIELD MEND.
                                ;
010650  022767  177777  167300          CMP     #-1,MEND        ;MEMORY END ADDR SET UP
010656  001003                         BNE     10$             ;YES
010660  016667  000004  167270          MOV     4(SP),MEND      ;SET END ADDR = MEMORY UPPER LIMIT
010666  026766  167260  000002  10$:    CMP     MSTRT,2(SP)     ;IS LOAD ADDRESS IN RANGE (LOW)
010674  103003                         BHIS    1$              ;OK, CONTINUE
010676  004767  002716                  JSR     PC,ERR19        ;OUT OF RANGE
010702  000432                         BR      BUFCX
010704  026766  167246  000004  1$:     CMP     MEND,4(SP)      ;IS ADDR IN RANGE (HIGH)
010712  101403                         BLOS    2$              ;YES, CONTINUE
010714  004767  002674                  JSR     PC,ERR20        ;OUT OF RANGE
010720  000423                         BR      BUFCX
010722  026767  167224  167226  2$:     CMP     MSTRT,MEND      ;IS START ADDR LOWER THAN END ADDR
010730  101403                         BLOS    3$              ;YES, CONTINUE
010732  004767  002656                  JSR     PC,ERR20        ;OUT OF RANGE
010736  000414                         BR      BUFCX           ;EXIT
                                ;
010740  012767  000001  167212  3$:     MOV     #1,INCVAL       ;BUFFERS INC BY 1
```

```
560 010746  012767  000003  167206          MOV.    #3,RSPCNT.      ;NUMBER.OF.SEGMENTS.(FOR.PROMPT)
561                                 ;
562.010754  011666  000004                  MOV.    (SP),4(SP)      ;MOVE.RETURN.ADDRESS.
563 010760  062706  000004                  ADD.    #4,SP.          ;ADJUST.SP.(FOR.MEM.LIMITS)
564 010764  000241                          CLC.
565 010766  000405                          BR      BUFXX.          ;RETURN.
566 010770  011666  000004          BUFCX:  MOV.    (SP),4(SP)
567 010774  062706  000004                  ADD.    #4,SP.
568 011000  000261                          SEC.
569 011002. 000207          BUFXX:  RTS.    PC.
```

```
                                ;
                                ;
                                ;       SET UP BEFORE PROMPTING ON ANY 'LD' COMMAND OR BEFORE
                                ;       EXECUTING ANY 'PR' COMMAND FOR NON-FSA MEMORIES.  THE
                                ;       FIELDS SET UP HERE ARE USED IN THE TMAIN PROMTPING
                                ;       SUBROUTINE PDATA AND THE PRINT ROUTINE PRDATA.
                                ;
                                ;       INPUT:
                                ;       2(SP)   LOAD/PRINT START ADDRESS
                                ;       4(SP)   LOWER MEMORY LIMIT
                                ;       6(SP)   UPPER MEMORY LIMIT
                                ;       APLACE  - TD CONTROL REGISTER BIT SETTING
                                ;
                                ;       OUTPUT:
                                ;       INCVAL  - VALUE BY WHICH MEMORY ADDRESS INCREMENTS
                                ;       MEND    - ADDRESS OF LAST WORD IN MEMORY TO BE LOADED/PRINTED
                                ;       RSPCNT  - NUMBER OF WORDS TO EXPECT IN COMMAND LINE (MEMORY WIDTH)
                                ;
                                ;       C-BIT CLEAR     NO ERROR (START ADDRESS IS IN RANGE)
                                ;       C-BIT SET       ERROR IN RANGE
                                ;
                                ;
                                ;       MEMORY          INCVAL          RSPCNT
                                ;       IB              2               1
                                ;       OB              2               1
                                ;       BT              1               1
                                ;
                                ;
000000 011004  012767  000002  167146  BUFS2::: MOV     #2,INCVAL       ;BUFFERS INC BY 2
000004 011012  000403                          BR      SET
000006 011014  012767  000001  167136  BUFSET:: MOV     #1,INCVAL       ;BUFFERS INC BY 1
                                ;
000016 011022  012767  000000  176376  SET:    MOV     #0,TD$SW        ;RESET TD
000024 011030  016767  166762  176360          MOV     APLACE,TD$CTW   ;SET TD CONTROL REG
                                ;
                                ;       IF THE FIELD MEND IS NOT =-1, THEN A 'LD' OR
                                ;       'PR' ROUTINE HAS PUT A MEMORY UPPER LIMIT IN
                                ;       THIS FIELD, EG: A COMMAND SUCH AS:
                                ;       >LD OB 0 10
                                ;       WOULD PUT 10 IN THE FIELD MEND.
                                ;
000036 011036  022767  177777  167112          CMP     #-1,MEND        ;MEMORY END ADDR SET UP
000044 011044  001003                          BNE     10$             ;YES
000046 011046  016667  000004  167102          MOV     4(SP),MEND      ;SET END ADDR = MEMORY UPPER LIMIT
000054 011054  026766  167072  000002  10$:    CMP     MSTRT,2(SP)     ;IS LOAD ADDRESS IN RANGE (LOW)
000062 011062  103003                          BHIS    1$              ;OK, CONTINUE
000064 011064  004767  002530                  JSR     PC,ERR19        ;OUT OF RANGE
000070 011070  000446                          BR      BUFCX2
000072 011072  026766  167060  000004  1$:     CMP     MEND,4(SP)      ;IS ADDR IN RANGE (HIGH)
000100 011100  101403                          BLOS    2$              ;YES, CONTINUE
000102 011102  004767  002506                  JSR     PC,ERR20        ;OUT OF RANGE
000106 011106  000437                          BR      BUFCX2
000110 011110  026767  167036  167040  2$:     CMP     MSTRT,MEND      ;IS START ADDR LOWER THAN END ADDR
000116 011116  101403                          BLOS    3$              ;YES, CONTINUE
000120 011120  004767  002470                  JSR     PC,ERR20        ;OUT OF RANGE
000124 011124  000430                          BR      BUFCX2          ;EXIT
                                ;
```

```
628 011126  032767  000001  167024  3$:     BIT     #1,INCVAL       ;IS MEMORY ADDR INCREMENT EVEN
629 011134  001013                          BNE     5$              ;NO, ANY START ADDR IS OK
630 011136  032767  000001  167006          BIT     #1,MSTRT        ;IS START ADDRESS EVEN
631 011144  001004                          BNE     4$              ;NO, ERROR
632 011146  032767  000001  167002          BIT     #1,MEND         ;IS END ADDRESS EVEN
633 011154  001403                          BEQ     5$              ;YES, CONTINUE
634 011156  004767  002416          4$:     JSR     PC,ERR23
635 011162  000411                          BR      BUFCX2
636                                 ;
637 011164  012767  000001  166770  5$:     MOV     #1,RSPCNT       ;NUMBER OF SEGMENTS (FOR PROMPT)
638 011172  011666  000004                  MOV     (SP),4(SP)      ;MOVE RETURN ADDRESS
639 011176  062706  000004                  ADD     #4,SP           ;ADJUST SP (FOR MEM LIMITS)
640 011202  000241                          CLC
641 011204  000405                          BR      BUFXX2          ;RETURN
642 011206  011666  000004          BUFCX2: MOV     (SP),4(SP)
643 011212  062706  000004                  ADD     #4,SP
644 011216  000261                          SEC
645 011220  000207                  BUFXX2: RTS     PC
```

```
                                        ;
                                        ;
                                        ;       LOAD·CONTROL FOR·FSA'S
                                        ;
                                        ;       INPUT:
                                        ;       INCVAL·  - VALUE·BY·WHICH·MEMORY·ADDRESS· INCREMENTS·
                                        ;       MSTR2·   - MEMORY·START ADDRESS·
                                        ;       MEND·    - MEMORY·END·ADDRESS·
                                        ;       DATA1, DATA2, DATA3  - SET·UP·BY ROUTINE·PDATA·
                                        ;       #RP·FLAG·  - DETERMINES WHETHER TO REPEAT PROMPT·
                                        ;       #LOOP·FLAG·  - DETERMINES·WHETHER TO LOOP ON LOAD·
                                        ;
                                        ;
011222                                  LDCNTL::
011222· 032767  000200  166564          BIT·    #RP,BASE·       ;REPEAT·PROMPT·
011230· 001424                          BEQ·    2$              ;NO, PROMPT·ONCE·
011232· 004767  000144                  1$:     JSR·    PC,PDATA·       ;READ·DATA·FROM·COMMAND·LINE
011236  103460                          BCS·    CNTLX·          ;ERROR·OR·END OF·MEMORY·
011240  102457                          BVS·    CNTLX·          ;<CR> RESPONSE·TO·PROMPT·
011242· 016767  166706  176362·         MOV·    MSTR2,TD$TAW    ;MOVE·ADDR·TO·TRANSFER·REG·
011250  016767  166724  176364          MOV·    DATA1,TD$TDW·   ;MOVE·SEG·1
011256  016767  166720  176364          MOV·    DATA2,TD$TDW    ;MOVE·SEG·2·
011264  016767  166714  176364          MOV·    DATA3,TD$TDW·   ;MOVE·SEG·3
011272· 066767  166662  166654          ADD·    INCVAL,MSTR2·   ;ADVANCE ADDRESS·
011300  000754                          BR      1$              ;REPEAT·
                                        ;
011302· 004767  000074                  2$:     JSR·    PC,PDATA·       ;PROMPT ONCE·
011306  103434                          BCS·    CNTLX·          ;END·OF·MEMORY OR·ERROR·
011310  102433                          BVS·    CNTLX·          ;<CR> RESPONSE·TO·PROMPT·
011312· 016767  166636  176362·  3$:    MOV·    MSTR2,TD$TAW    ;SET UP TRANNSFER·ADDRESS
011320  016767  166654  176364          MOV·    DATA1,TD$TDW·   ;MOVE·SEG·1
011326  016767  166650  176364          MOV·    DATA2,TD$TDW    ;MOVE·SEG·2·
011334  016767  166644  176364          MOV·    DATA3,TD$TDW·   ;MOVE·SEG·3
011342· 066767  166612  166604          ADD·    INCVAL,MSTR2·   ;ADVANCE ADDRESS·
011350  026767  166600  166600          CMP·    MSTR2,MEND·     ;HAS·UPPER·MEMORY·LIMIT·BEEN REACHED·
011356· 101755                          BLOS·   3$              ;NO, CONTINUE·
011360  032767  000004  166426          BIT·    #LOOP,BASE·     ;LOOP·ON ?·
011366  001404                          BEQ·    CNTLX·          ;NO, EXIT·
011370  016767  166556  166556          MOV·    MSTRT,MSTR2·    ;REINITIALIZE·ADDRESS·
011376  000745                          BR      3$
                                        ;
011400                                  CNTLX:
011400  000207                          RTS·    PC·
```

```
                                ;
                                ;
                                ;       PROMPTING CONTROL FOR LOADING ALL MEMORIES AND BUFFERS,
                                ;       READ NUMERIC DATA FROM THE COMMAND LINE, CONVERT AND STORE
                                ;       INTO A COMMON BUFFER, EG, IF THE INITIATING COMMAND IS:
                                ;       >LD AM 0
                                ;       THIS ROUTINE WILL PUT OUT A PROMPT AND EXPECT 3 NUMERIC
                                ;       VALUES IN RETURN:
                                ;       >000000 000000 000000
                                ;       THIS ROUTINE CONVERTS THESE ASCII OCTAL VALUES INTO BINARY
                                ;       AND STORES THEM INTO DATA1, DATA2, DATA3
                                ;
                                ;       INPUT: (SET UP BY TMAIN SUBROUTINE BUFSET)
                                ;       MSTR2   - CURRENT MEMORY ADDRESS
                                ;       MEND    - MEMORY UPPER ADDRESS LIMIT
                                ;       RSPCNT  - NUMBER OF WORDS TO EXPECT IN COMMAND LINE
                                ;
                                ;       OUTPUT:
                                ;       WWORDS (DATA1, DATA2, DATA3) DEPENDING UPON RSPCNT
                                ;
                                ;       C-BIT CLEAR, V-BIT CLEAR        NORMAL RETURN
                                ;       C-BIT CLEAR, V-BIT SET          <CR> RESPONSE TO PROMPT
                                ;       C-BIT SET, V-BIT CLEAR          END OF MEMORY OR CONVERSION ERROR
                                ;
                                ;       REGISTERS 1, 4, 5 DESTROYED
                                ;
                                ;
011402                          PDATA::
011402  026767  166546  166546            CMP     MSTR2,MEND      ;UPPER MEMORY LIMIT REACHED
011410  101403                            BLOS    10$             ;NO, CONTINUE
011412  004767  002146                    JSR     PC,ENDMEM       ;END OF MEMORY REACHED
011416  000470                            BR      PDCX            ;SET CARRY AND EXIT
                                ;
011420  016701  166530            10$:    MOV     MSTR2,R1        ;PREPARE TO PRINT ADDRESS
011424  012705  005654'                   MOV     #PRINT,R5       ;POINT TO PRINT LINE
011430  004767  001672                    JSR     PC,UNPK         ;CONVERT ADDRESS
011434  004767  001732                    JSR     PC,CONSOL       ;PRINT OUT ADDRESS
                                ;
011440  012767  020040  176326            MOV     #20040,GCMBLK+G.DPRM ;ERASE CR+LF
011446  004767  001036                    JSR     PC,GCONLY       ;PROMPT
011452  112767  000015  176314            MOVB    #15,GCMBLK+G.DPRM ;RESTORE CR
011460  112767  000012  176307            MOVB    #12,GCMBLK+G.DPRM+1 ;RESTORE LF
011466  004767  001406                    JSR     PC,FIND         ;LOCATE FIRST DATA WORD IN COMMAND LINE
011472  103437                            BCS     PDVX            ;<CR> RESPONSE, EXIT
                                ;
011474  012705  000200'                   MOV     #WWORDS,R5      ;WORDS FROM COMMAND LINE GO INTO THIS TABLE
011500  016704  166456                    MOV     RSPCNT,R4       ;NUMBER OF WORDS TO EXPECT
011504  004767  001506            2$:     JSR     PC,PACK         ;CONVERT WORD TO BINARY
011510  103003                            BCC     3$              ;OK, CONTINUE
011512  004767  002176                    JSR     PC,ERR4         ;'INVALID NUMERIC VALUE'
011516  000430                            BR      PDCX            ;AND EXIT
011520  016725  166266            3$:     MOV     BINWD,(R5)+     ;MOVE WORD TO TABLE
011524  005304                            DEC     R4              ;SUB FROM LOOP COUNT
011526  001427                            BEQ     PDCCX           ;FINISHED
011530  004767  001344                    JSR     PC,FIND         ;FIND NEXT WORD
011534  103003                            BCC     4$              ;OK, CONTINUE
011536  004767  002156                    JSR     PC,ERR3         ;'MISSING OPERAND'
```

```
748 011542  000416                        BR      PDCX
749                                 ;
750 011544  122711  000122      4$:     CMPB    #'R,(R1)        ;REPEAT PATTERN (FSA'S ONLY)
751 011550  001355                      BNE     2$              ;NO
752 011552  016725  166234              MOV     BINWD,(R5)+     ;WRITE SAME PATTERN TO NEXT SEG
753 011556  022704  000001              CMP     #1,R4           ;ALREADY FILLED LAST SEG
754 011562  001411                      BEQ     PDCCX           ;YES, EXIT
755 011564  016725  166222              MOV     BINWD,(R5)+
756 011570  000406                      BR      PDCCX           ;AND EXIT
757                                 ;
758 011572  000241              PDVX:   CLC
759 011574  000262                      SEV                     ;INDICATE <CR> RESPONSE
760 011576  000405                      BR      PDDX
761 011600  000242              PDCX:   CLV
762 011602  000261                      SEC                     ;INDICATE END OF MEMORY REACHED
763 011604  000402                      BR      PDDX
764 011606  000241              PDCCX:  CLC
765 011610  000242                      CLV
766 011612  000207              PDDX:   RTS     PC
```

```
                                        ;
                                        ;
                                        ;       PRINT·CONTROL·FOR·FSA'S·
                                        ;
                                        ;       FIELDS·ALREADY·SET·UP·BY·'BUFFSA':
                                        ;       INPUT:
                                        ;       MSTR2·    - MEMORY·ADDRESS·
                                        ;       MEND·     - MEMORY·END·ADDRESS·
                                        ;       INCVAL·   - VALUE·BY·WHICH·MEMORY·ADDRESS·INCREMENTS·
                                        ;       #LOOP·FLAG·  - REPEAT·MEMORY·READ
                                        ;
                                        ;       OUTPUT:
                                        ;       DATA1, DATA2, DATA3 (INPUT·TO·ROUTINE·PRDATA)
                                        ;
                                        ;
011614                                  PRCNTL::
011614  016767  166334  176362·  1$:     MOV·    MSTR2,TD$TAW    ;MOVE·ADDR·TO·TRANSFER·REG·
011622· 016767  176374  166350          MOV·    TD$TDR,DATA1    ;MOVE·SEG·1
011630  016767  176374  166344          MOV·    TD$TDR,DATA2·   ;MOVE·SEG·2·
011636  016767  176374  166340          MOV·    TD$TDR,DATA3    ;MOVE·SEG·3
011644  004767  000040                  JSR·    PC,PRDATA·      ;PRINT·MEMORY·CONTENTS·
                                        ;
011650  066767  166304  166276          ADD·    INCVAL,MSTR2·   ;ADVANCE ADDRESS·
011656  026767  166272  166272·         CMP·    MSTR2,MEND·     ;HAS·UPPER·MEMORY·LIMIT·BEEN·REACHED·
011664  101753                          BLOS·   1$              ;NO, CONTINUE·
011666  032767  000004  166120          BIT·    #LOOP,BASE·     ;LOOP·
011674  001404                          BEQ·    PCNTLX·         ;NO, EXIT·
011676  016767  166250  166250          MOV·    MSTRT,MSTR2·    ;REINIT·START·ADDRESS·
011704  000743                          BR      1$              ;AND·REPEAT·
                                        ;
011706                                  PCNTLX:
011706  000207                          RTS·    PC·
```

```
                                ;
                                ;
                                ;       COMMON PRINT ROUTINE
                                ;       THIS ROUTINE CONVERTS VALUES IN THE FIELD 'WWORDS' TO
                                ;       ASCII OCTAL (DEPENDING ON RSPCNT) AND PRINTS THEM (TT0)
                                ;
                                ;       INPUT:
                                ;       #OUT FLAG       - WHEN LOOP OPTION IS IN EFFECT, PRINT 1 WORD ONLY
                                ;       #ONCE FLAG      - CONTROL PRINTING OF 1 WORD WHEN #OUT IS SET
                                ;       RSPCNT          - NUMBER OF WORDS TO PRINT (SEE 'BUFSET' RTN)
                                ;       MSTR2           - MEMORY ADDRESS (SEE 'BUFSET')
                                ;       WWORDS          - MEMORY WORDS (DATA1, DATA2, DATA3, DATA4)
                                ;
                                ;       REGISTERS 1, 3, 4, 5 DESTROYED
                                ;
                                ;
011710                          PRDATA::
011710  032767  002000  166076          BIT     #OUT,BASE       ;OUTPUT CONTROL ON
011716  001407                        BEQ     10$             ;NO, SKIP OUTPUT CONTROL
011720  032767  000010  166066        BIT     #ONCE,BASE      ;ONE LINE PRINTED
011726  001026                        BNE     PRDX            ;YES, EXIT
011730  052767  000010  166056        BIS     #ONCE,BASE      ;SET FLAG FOR NEXT TIME
                                ;
011736  016701  166212          10$:  MOV     MSTR2,R1        ;FIRST CONVERT ADDRESS TO ASCII
011742  012705  005654'              MOV     #PRINT,R5       ;POINT TO PRINT LINE
011746  004767  001354                JSR     PC,UNPK         ;PERFORM CONVERSION
                                ;
011752  012704  000200'               MOV     #WWORDS,R4      ;POINT TO WORDS FOR PRINTING
011756  016703  166200                MOV     RSPCNT,R3       ;NUMBER OF MEM WORDS TO PRINT
011762  062705  000002          1$:   ADD     #2,R5           ;ADVANCE PRINT LINE POINTER
011766  012401                        MOV     (R4)+,R1        ;LOAD ONE WORD
011770  004767  001332                JSR     PC,UNPK         ;CONVERT IT TO ASCII
011774  005303                        DEC     R3              ;DEC WORD COUNT
011776  001371                        BNE     1$              ;REPEAT
012000  004767  001366                JSR     PC,CONSOL       ;PRINT LINE
                                ;
012004  000207                  PRDX: RTS     PC
```

```
                                ;
                                ;
                                ;       CALCULATE 8-BIT BYTE OFFSET INTO 6-BIT BYTE TABLE.
                                ;
                                ;       INPUT:
                                ;       BINWD   - 'CHANGE' OR 'END' VALUE FROM COMMAND LINE
                                ;                 (= OFFSET INTO 6-BIT BYTE TABLE)
                                ;
                                ;       OUTPUT:
                                ;       OFF6    -  8-BIT BYTE OFFSET INTO 6-BIT BYTE TABLE
                                ;       COUNT   -  FOUR 6-BIT BYTES FIT IN THREE 8-BIT
                                ;                  BYTES. 'COUNT' REPRESENTS THE POSITION
                                ;                  OF A 6-BIT BYTE WITHIN THE THREE 8-BIT
                                ;                  BYTES. POSITIONS ARE 0, 1, 2, 3.
                                ;
                                ;       TO GET THE OFFSET VALUE, MULTIPLY THE BINWD (6-BIT
                                ;       LOCATION) VALUE BY 3, THEN DIVIDE BY 4. TO GET THE
                                ;       POSITION, SUBTRACT THE REMAINDER FROM THE DIVISION
                                ;       FROM 4.
                                ;
                                ;
012006                          OFFSET::
012006  005767  166000                  TST     BINWD           ;CHECK FOR VALUE = 0
012012  001005                          BNE     1$              ;NO, CONTINUE
012014  005067  166154                  CLR     OFF6            ;START AT FIRST BYTE
012020  005067  166142                  CLR     COUNT           ;START AT FIRST BIT POSITION
012024  000422                          BR      2$
                                ;
012026  016701  165760          1$:     MOV     BINWD,R1        ;PREPARE FOR MULT
012032  012700  000003                  MOV     #3.,R0          ;MULTIPLIER
012036  004767  000000G                 JSR     PC,$MUL         ;MULT VALUE BY 6
012042  010100                          MOV     R1,R0           ;MOVE PRODUCT TO R0 FOR DIVIDE
012044  012701  000004                  MOV     #4.,R1          ;DIVISOR IN R1
012050  004767  000000G                 JSR     PC,$DIV         ;GET NUMBBER OF 8-BIT BYTES
012054  010067  166114                  MOV     R0,OFF6         ;SAVE OFFSET INTO TABLE
012060  012702  000004                  MOV     #4.,R2          ;PREPARE TO CALCULATE POSITION
012064  160102                          SUB     R1,R2           ;SUB REMAINDER FROM ABOVE DIVISION
012066  010267  166074                  MOV     R2,COUNT        ;SAVE POSITION
012072  000207                  2$:     RTS     PC
```

```
                                ;
                                ;
                                ;       BUILD 6-BIT BYTE 'DATA' BUFFER
                                ;       PROCESS ONE COMMAND LINE'S WORTH OF 8-BIT BYTES,
                                ;       CONVERT THEM TO 6-BIT BYTES AND ADD THEM TO THE
                                ;       6-BIT BYTE TABLE, ALSO SAVE THE 8-BIT BYTES IN
                                ;       THEIR CURRENT FORM FOR USE BY 'LI' COMMAND.
                                ;
                                ;       INPUT:
                                ;       GCMLEN  - LENGTH OF COMMAND LINE
                                ;       COUNT   - 6-BIT BYTE POSITION (SEE ROUTINE 'OFFSET')
                                ;       R1      - POINTER TO COMMAND LINE
                                ;       R4      - POINTER TO 6-BIT BYTE TABLE
                                ;       R5      - POINTER TO 8-BIT ASCII MIRROR IMAGE OF COMMAND
                                ;                 LINE INPUT TABLE
                                ;
                                ;       OUTPUT:
                                ;       DPLUS   - NUMBER OF 6-BIT CHARACTERS ADDED TO 6-BIT
                                ;                 BYTE TABLE IN CURRENT CALL OF DBLD.
                                ;
                                ;
012074                          DBLD::
012074  016767  166042  166070          MOV     GCMLEN,DPLUS    ;NUMBER OF CHARS IN COMMAND LINE
                                ;
                                ;       THE STRINGS <12> AND <15> ARE ALLOWED IN THE COMMAND LINE
                                ;       FOR THE ENTERING OF CARRAIGE RETURNS AND LINE FEEDS, A
                                ;       CARRIAGE RETURN IS SAVED IN THE MIRROR TABLE AS A BINARY
                                ;       15, A LINE FEED AS A BINARY 12, THE BRACKETS ARE RE-INSTATED
                                ;       BY THE 'LI' COMMAND FOR PRINTING.
                                ;
012102  121127  000074          DBLD1:  CMPB    (R1),#'<        ;LOOK FOR <15> OR <12>
012106  001047                  BNE     NOCRLF          ;NO ADJUSTMENT NEEDED
012110  126127  000003  000076          CMPB    3(R1),#'>       ;ARE BRACKETS CLOSED
012116  001043                  BNE     NOCRLF          ;NO, TREAT BRACKET AS CHAR (INVALID)
012120  126127  000001  000061          CMPB    1(R1),#'1       ;FIRST CHAR OF 12 OR 15
012126  001403                  BEQ     1$              ;OK, CONTINUE
012130  004767  001454          JSR     PC,ERR21        ;ILLEGAL 'DATA' CHARACTER
012134  000557                  BR      DBLDSX
                                ;
                                ;       PARSE THE SECOND OF THE 2 CHARS BETWEEN THE BRACKETS
                                ;
012136  012702  000010          1$:     MOV     #10,R2          ;START WITH BINARY 10
012142  126127  000002  000062          CMPB    2(R1),#'2       ;LINE FEED
012150  001003                  BNE     2$              ;NO, TRY CR
012152  062702  000002                  ADD     #2,R2           ;MAKE R2 = 12.
012156  000411                  BR      4$
012160  126127  000002  000065  2$:     CMPB    2(R1),#'5       ;CARRAIGE RETURN
012166  001403                  BEQ     3$              ;OK, CONTINUE
012170  004767  001414          JSR     PC,ERR21        ;ILLEGAL 'DATA' CHARACTER
012174  000537                  BR      DBLDSX
                                ;
                                ;       MOVE BINARY VALUE TO MIRROR TABLE
                                ;
012176  062702  000005          3$:     ADD     #5,R2           ;MAKE R2 = 15.
012202  110225                  4$:     MOVB    R2,(R5)+        ;MOVE CR OR LF INTO MIRROR TABLE
012204  062701  000004                  ADD     #4,R1           ;ADJUST COMMAND LINE POINTER
012210  162767  000003  165724          SUB     #3,GCMLEN       ;ADJUST COMMAND LINE CHAR COUNT
```

```
 936 012216  162767  000003  165746          SUB·    #3,DPLUS·       ;ADJUST·'DATA' CHAR·COUNT·FOR·THIS LINE·
 937 012224  000412·                         BR      MOVTR·
 938                                 ;
 939                                 ;       MOVE·CHARACTER·FROM·COMMAND·LINE TO·MIRROR TABLE·
 940                                 ;
 941 012226                          NOCRLF:
 942·012226  111125                          MOVB    (R1),(R5)+      ;MOVE·CHAR·TO·ASCII SAVE·BUFFER·
 943 012230  022705  005046'                 CMP·    #DSEND,R5       ;END·OF·BUFFER·REACHED·
 944 012234  002005                          BGE·    1$              ;NO, CONTINUE·
 945 012236  004767  001322                  JSR·    PC,ENDMEM·      ;ISSUE·MESSAGE·
 946 012242· 105067  172600·                 CLRB    DSEND·          ;CLEAR·LAST BYTE·OF·TABLE·
 947 012246  000512·                         BR      DBLDSX·         ;AND·EXIT·
 948 012250  112102·                 1$:     MOVB    (R1)+,R2·       ;LOAD CHAR·FROM COMMAND·LINE
 949                                 ;
 950                                 ;       TRANSLATE·CHARACTER·FROM·COMMAND LINE·INTO 6-BIT·EQUIVALENT·
 951                                 ;
 952·012252·                         MOVTR:
 953 012252· 116202· 005050'                 MOVB    TR6TBL(R2),R2·  ;LOAD·6-BIT·CHAR·
 954 012256  002003                          BGE·    10$             ;VALID·CHAR, CONTINUE·
 955 012260  004767  001324·                 JSR·    PC,ERR21        ;ILLEGAL 'DATA' CHARACTER·
 956 012264  000503                          BR      DBLDSX·
 957                                 ;
 958                                 ;       PUT·6-BIT·CHARACTER·INTO·TABLE·
 959                                 ;       TAKE·INTO·ACCOUNT·'POSITION', 4 6-BIT·BYTES·FIT·INTO
 960                                 ;       3 8-BIT·BYTES. POSITIONS·ARE·0, 1, 2, 3.
 961                                 ;       POSITION:
 962·                                ;       0       00000000 00000000 00111111
 963                                 ;       1       00000000 00001111 11000000
 964                                 ;       2·      00000011 11110000 00000000
 965                                 ;       3       11111100 00000000 00000000
 966                                 ;
 967                                 ;       POSITION·0
 968                                 ;
 969 012266  010203                  10$:    MOV·    R2,R3           ;COPY·FOR·LATER·USE·
 970 012270  005767  165672                  TST·    COUNT·          ;POSITION·=·0
 971 012274  001004                          BNE·    1$              ;NO, TRY 1
 972·012276  142714  000077·                 BICB·   #077,(R4)       ;CLEAR·AHEAD·OF·BIT·SET·
 973 012302· 150314                          BISB·   R3,(R4)         ;'OR' IN 6 BITS·
 974 012304  000464                          BR      NEXTD·          ;GET·NEXT·CHAR·
 975                                 ;
 976                                 ;       POSITION·1
 977                                 ;
 978 012306  122767  000001  165652· 1$:     CMPB·   #1,COUNT·       ;POSITION·=·1
 979 012314  001022·                         BNE·    2$              ;NO, TRY 2·
 980 012316  142703  000074·                 BICB·   #074,R3         ;CLEAR·TOP·4 BITS·OF·6-BIT·CHAR·(00111100)
 981 012322· 006303·                         ASL·    R3              ;SHIFT·2 BITS·LEFT 6
 982·012324  006303                          ASL·    R3
 983 012326  006303                          ASL·    R3
 984 012330  006303                          ASL·    R3
 985 012332· 006303                          ASL·    R3
 986 012334  006303                          ASL·    R3
 987 012336  142714  000300·                 BICB·   #300,(R4)       ;CLEAR·AHEAD·OF·BIT·SET·
 988 012342· 150324                          BISB·   R3,(R4)+        ;'OR' IN 6-BIT CHAR·AND·INCREMENT
 989                                 ;
 990 012344  010203                          MOV·    R2,R3           ;LOAD ORIGINAL·CHAR·
 991 012346  006203                          ASR·    R3              ;LEAVE·4 BITS·OF·6-BIT·CHAR·
 992·012350  006203                          ASR·    R3
```

```
 993 012352  142714  000017                  BICB    #017,(R4)       ;CLEAR AHEAD OF BIT SET
 994 012356  150314                          BISB    R3,(R4)         ;'OR' IN 6-BIT CHAR
 995 012360  000436                          BR      NEXTD           ;GET NEXT COMMAND LINE CHAR
 996                                ;
 997                                ;       POSITION 2
 998                                ;
 999 012362  022767  000002  165576  2$:     CMP     #2,COUNT        ;IS PASS COUNT = 2
1000 012370  001022                          BNE     3$              ;NO, TRY 3
1001 012372  142703  000060                  BICB    #060,R3         ;CLEAR 2 BITS (00110000)
1002 012376  006303                          ASL     R3              ;SHIFT 4 REMAINING BITS LEFT 4
1003 012400  006303                          ASL     R3
1004 012402  006303                          ASL     R3
1005 012404  006303                          ASL     R3
1006 012406  142714  000360                  BICB    #360,(R4)       ;CLEAR AHEAD OF BIT SET
1007 012412  150324                          BISB    R3,(R4)+        ;'OR' IN PART OF A 6-BIT CHAR
1008                               ;
1009 012414  010203                          MOV     R2,R3           ;LOAD ORIGINAL CHAR
1010 012416  006203                          ASR     R3              ;SHIFT 4 BITS RIGHT 4
1011 012420  006203                          ASR     R3
1012 012422  006203                          ASR     R3
1013 012424  006203                          ASR     R3
1014 012426  142714  000003                  BICB    #3,(R4)         ;CLEAR AHEAD
1015 012432  150314                          BISB    R3,(R4)         ;'OR' IN PART OF 6-BIT CHAR
1016 012434  000410                          BR      NEXTD
1017                               ;
1018                               ;       POSITION 3
1019                               ;
1020 012436  006303                 3$:     ASL     R3              ;SHIFT 6 BITS LEFT 2
1021 012440  006303                          ASL     R3
1022 012442  142714  000374                  BICB    #374,(R4)       ;CLEAR AHEAD
1023 012446  150324                          BISB    R3,(R4)+        ;'OR' IN 6 BITS
1024 012450  012767  177777  165510          MOV     #-1,COUNT       ;COUNT WILL BE RESET TO ZERO
1025                               ;
1026 012456  005267  165504         NEXTD:  INC     COUNT           ;BUMP 6-BIT BYTE COUNTER
1027 012462  005367  165454                  DEC     GCMLEN          ;DEC BYTE COUNT
1028 012466  001406                          BEQ     DBLDCX
1029 012470  000167  177406                  JMP     DBLD1           ;PROCESS NEXT INPUT CHARACTER
1030                               ;
1031 012474  005067  165470         DBLDSX: CLR     DATALN          ;AS THOUGH RTN WAS NOT ENTERED
1032 012500  000261                          SEC
1033 012502  000401                          BR      DBLDXX          ;AND EXIT
1034 012504  000241                 DBLDCX: CLC
1035 012506  000207                 DBLDXX: RTS     PC
```

```
1037                                    ;
1038                                    ;
1039                                    ;       GET COMMAND LINE FROM TERMINAL
1040                                    ;
1041                                    ;       OUTPUT:
1042                                    ;       GCMBUF  - WORK AREA TO HOLD COMMAND LINE
1043                                    ;       GCMLEN  - LENGTH OF LINE READ
1044                                    ;       GCMPNT  - POINTER TO COMMAND LINE (SET TO POINT TO BEGINNING)
1045                                    ;
1046                                    ;       SEE 'FIND' SUBROUTINE FOR THE USE AND UPDATING OF THESE FIELDS.
1047                                    ;
1048                                    ;
1049 012510                            GCONLY::
1050 012510  012700  000020'                   MOV     #GCMBUF,R0      ;POINT TO COMMAND LINE BUFFER
1051 012514  012701  000120                    MOV     #80.,R1         ;NUMBER OF BYTES IN BUFFER
1052 012520  112720  000040            1$:     MOVB    #40.,(R0)+      ;CLEAR COMMAND LINE
1053 012524  005301                            DEC     R1              ;FINISHED?
1054 012526  001374                            BNE     1$              ;NO
1055                                    ;
1056 012530                                    GCML$   #GCMBLK
     012530  012700  007614'                   MOV     #GCMBLK,R0
     012534  005060  000142                    CLR     G.PSDS(R0)
     012540  004767  000000G                   JSR     PC,.GCML1
1057 012544  016067  000146  165370           MOV     G.CMLD(R0),GCMLEN       ;SAVE LENGTH
1058 012552  012767  000020' 165364           MOV     #GCMBUF,GCMPNT  ;INIT COMMAND LINE POINTER
1059 012560  012701  000020'                   MOV     #GCMBUF,R1      ;POINT R1 TO COMMAND LINE
1060 012564  000207                            RTS     PC
```

```
                                        ;
                                        ;
                                        ;       PROMPT FOR LOOP ON TEST.
                                        ;       GIVE DIRECTIONS FOR STOPPING LOOP (AND HANG QIO)
                                        ;
                                        ;
012566                                  LOOPR::
012566  004767  000306                          JSR     PC,FIND         ;LOOK FOR LOOP OPTION CHAR IN COMMAND LINE
012572  103444                          BCS     LOOPX           ;NOTHING THERE, EXIT
012574  122711  000114          1$:     CMPB    #'L,(R1)        ;LOOP INDICATOR ?
012600  001410                          BEQ     HANG            ;YES, SET LOOP FLAG
012602  004767  001122                  JSR     PC,ERR1         ;LOOP OPTION ERROR
012606  004767  001122                  JSR     PC,LPTST        ;PROMPT FOR LOOP OPTION
012612  004767  000262                  JSR     PC,FIND         ;ANY RESPONSE
012616  103432                          BCS     LOOPX           ;NO LOOP AFTER ALL
012620  000765                          BR      1$              ;CHECK RESPONSE
                                        ;
012622  052767  000004  165164  HANG::  BIS     #LOOP,BASE      ;SET FLAG FOR LOOP
012630  052767  000020  165156          BIS     #ASTFLG,BASE    ;SET FLAG FOR QIO ISSUED
012636  004767  000726                  JSR     PC,STOP         ;GIVE DIRECTIONS FOR STOPPING TEST
                                        ;
012642                                  QIO$S   #IO.ATA,#LUN.TT,,,,,<#AST>
012642  005046                          CLR     -(SP)
012644  005046                          CLR     -(SP)
012646  005046                          CLR     -(SP)
012650  005046                          CLR     -(SP)
012652  005046                          CLR     -(SP)
012654  012746  012766'                 MOV     #AST,-(SP)
012660  005046                          CLR     -(SP)
012662  005046                          CLR     -(SP)
012664  005046                          CLR     -(SP)
012666  012746  000001                  MOV     #LUN.TT,-(SP)
012672  012746  000000G                 MOV     #IO.ATA,-(SP)
012676  012746                          MOV     (PC)+,-(SP)
012700     001     014                  .BYTE   1,12.
012702  104377                          EMT     ^O<377>
                                        ;
012704                                  LOOPX:
012704  000207                          RTS     PC
                                        ;
                                        ;
                                        ;
                                        ;       KILL AST
                                        ;
                                        ;
012706                                  KILL::
012706  032767  000020  165100          BIT     #ASTFLG,BASE    ;WAS AST ISSUED
012714  001423                          BEQ     1$
012716  042767  000020  165070          BIC     #ASTFLG,BASE
012724                                  QIOW$S  #IO.DET,#LUN.TT
012724  005046                          CLR     -(SP)
012726  005046                          CLR     -(SP)
012730  005046                          CLR     -(SP)
012732  005046                          CLR     -(SP)
012734  005046                          CLR     -(SP)
012736  005046                          CLR     -(SP)
012740  005046                          CLR     -(SP)
                                        CLR     -(SP)
```

```
        012742  005046                          CLR     -(SP)
        012744  005046                          CLR     -(SP)
        012746  012746  000001                  MOV     #LUN.TT,-(SP)
        012752  012746  000000G                 MOV     #IO.DET,-(SP)
        012756  012746                          MOV     (PC)+,-(SP)
        012760     003     014                  .BYTE   3,12.
        012762  104377                          EMT     ^O<377>
  1098  012764  000207                  1$:     RTS     PC
  1099                                          ;
  1100                                          ;
  1101                                          ;       AST
  1102                                          ;
  1103                                          ;
  1104  012766                          AST:
  1105  012766  012667  165154                  MOV     (SP)+,ASTWRD    ;GET CHAR OFF STACK
  1106  012772  042767  000004  165014          BIC     #LOOP,BASE      ;CLEAR LOOP FLAG
  1107  013000                                  ASTX$S
        013000  012746                          MOV     (PC)+,-(SP)
        013002     163     001                  .BYTE   115.,1
        013004  104377                          EMT     ^O<377>
```

```
1109                                    ;
1110                                    ;
1111                                    ;       SCAN A TABLE FOR A VALID COMMAND/MNEMONIC
1112                                    ;
1113                                    ;       INPUT:
1114                                    ;       R0 = NUMBER OF ENTRIES IN COMMAND TABLE
1115                                    ;       R1 -> CHAR STRING IN GCML COMMAND LINE
1116                                    ;       R2 -> TOP OF COMMAND TABLE
1117                                    ;
1118                                    ;       OUTPUT:
1119                                    ;       R1 -> ROUTINE THAT GOVERNS THE COMMAND (IF MATCH WAS MADE)
1120                                    ;       R1 -> CHAR STRING IN COMMAND LINE (IF NO MATCH WAS MADE)
1121                                    ;       R0 = RELATIVE POSITION OF MATCHED ENTRY IN TABLE
1122                                    ;
1123                                    ;
1124 013006                            SCAN::
1125 013006  010346                             MOV     R3,-(SP)        ;SAVE R3
1126 013010  010046                             MOV     R0,-(SP)        ;SAVE # ENTRIES
1127 013012  010146                             MOV     R1,-(SP)        ;SAVE POINTER TO BEGINNING OF STRING
1128                                    ;
1129 013014  011601                     FNOUT1: MOV     (SP),R1         ;POINT TO NON-BLANK IN COMMAND LINE
1130 013016  012703  000002                     MOV     #2,R3           ;NUMBER OF CHARS IN NON-BLANK FIELD
1131 013022  122122                     FNIN1:  CMPB    (R1)+,(R2)+     ;DOES COMMAND LINE MATCH TABLE ENTRY
1132 013024  001003                             BNE     FNOUT2          ;NO, TRY NEXT TABLE ENTRY
1133 013026  005303                             DEC     R3              ;SUB FROM LOOP COUNT
1134 013030  001374                             BNE     FNIN1
1135 013032  000411                             BR      FNMTCH          ;COMMAND FOUND IN TABLE
1136 013034  060302                     FNOUT2: ADD     R3,R2           ;ADD # UNCOMPARED CHARS TO POINTER
1137 013036  005202                             INC     R2              ;THEN ADJUST TO NEXT TABLE ENTRY
1138 013040  005300                             DEC     R0              ;SUB FROM OUTER LOOP COUNT
1139 013042  001364                             BNE     FNOUT1          ;TRY AGAIN
1140 013044  012601                             MOV     (SP)+,R1        ;RESTORE COMMAND LINE POINTER
1141 013046  012600                             MOV     (SP)+,R0
1142 013050  012603                             MOV     (SP)+,R3        ;RESTORE R3
1143 013052  000261                             SEC                     ;COMMAND NOT IN TABLE
1144 013054  000207                             RTS     PC
1145                                    ;
1146 013056  010201                     FNMTCH: MOV     R2,R1           ;POINT R1 AT RTN ADDR IN TABLE
1147 013060  062706  000002                     ADD     #2,SP           ;POINT TO INCOMING R0 ON STACK
1148 013064  012602                             MOV     (SP)+,R2        ;GET TOTAL # TABLE ENTRIES
1149 013066  160002                             SUB     R0,R2           ;GET POSITION OF MATCHED ENTRY
1150 013070  010200                             MOV     R2,R0           ;PUT IN R0 FOR RETURN
1151 013072  012603                             MOV     (SP)+,R3        ;RESTORE R3
1152 013074  000241                             CLC
1153 013076  000207                             RTS     PC
```

```
                                ;
                                ;
                                ;       FIND THE NEXT NON-BLANK IN THE COMMAND BUFFER,
                                ;       THEN FIND THE LENGTH OF THE STRING THAT STARTS WITH THAT CHARACTER.
                                ;
                                ;       INPUT:
                                ;       GCMLEN  -  NUMBER OF UNPROCESSED BYTES IN COMMAND LINE
                                ;       GCMPNT  - ADDR OF NEXT UNPROCESSED POSITION IN COMMAND LINE
                                ;
                                ;       OUTPUT:
                                ;       R1 -> STRING, R0 = LENGTH OF STRING
                                ;       GCMLEN, GCMPNT UPDATED FOR NEXT ENTRY INTO THIS ROUTINE
                                ;
                                ;       THIS ROUTINE IS DESIGNED TO BE ENTERED A NUMBER OF TIMES
                                ;       IN THE PARSING OF A COMMAND LINE.  THE FIELDS GCMLEN AND
                                ;       GCMPNT ARE REFRESHED WHEN A NEW COMMAND LINE IS READ
                                ;       (SEE THE SUBROUTINE 'GCONLY').
                                ;
                                ;
013100                          FIND::
010246                                  MOV     R2,-(SP)        ;SAVE R2
016701  165034                          MOV     GCMLEN,R1       ;# BYTES REMAINING IN COMMAND BUFFER
001440                                  BEQ     FSECX           ;THERE ARE NONE
016702  165030                          MOV     GCMPNT,R2       ;LOAD CURRENT POINTER
122712  000040                  1$:     CMPB    #40,(R2)        ;LOOK FOR A BLANK
001403                                  BEQ     10$             ;OK, BUMP TO NEXT CHAR
122712  000054                          CMPB    #',,(R2)        ;COMMA IN COMMAND LINE
001004                                  BNE     2$              ;TREAT COMMA AS BLANK
005202                          10$:    INC     R2              ;BUMP POINTER
005301                                  DEC     R1              ;SUB FROM REMAINING LENGTH
001367                                  BNE     1$
000424                                  BR      FSECX           ;NO NON-BLANK FOUND
                                ;
010246                          2$:     MOV     R2,-(SP)        ;TEMP SAVE POINTER TO BEGINNING OF STRING
005000                                  CLR     R0              ;CLEAR CHAR COUNT
122712  000040                  3$:     CMPB    #40,(R2)        ;LOOK FOR A BLANK
001407                                  BEQ     4$              ;FOUND END OF STRING
122712  000054                          CMPB    #',,(R2)        ;TREAT COMMAS AS BLANKS
001404                                  BEQ     4$
005202                                  INC     R2              ;BUMP POINTER
005200                                  INC     R0              ;BUMP CHAR COUNT
005301                                  DEC     R1              ;SUB FROM BYTES REMAINING
001366                                  BNE     3$
                                ;
010267  164750                  4$:     MOV     R2,GCMPNT       ;SAVE POINTER FOR NEXT TIME
010167  164742                          MOV     R1,GCMLEN       ;SAVE BYTES REMAINING FOR NEXT TIME
012601                                  MOV     (SP)+,R1        ;POINTER TO BEGINNING OF STRING
012602                                  MOV     (SP)+,R2        ;RESTORE R2
000241                                  CLC
000207                                  RTS     PC
                                ;
012602                          FSECX:  MOV     (SP)+,R2        ;RESTORE R2
000261                                  SEC
000207                                  RTS     PC
```

```
                                ;
                                ;
                                ;       CONVERT A NUMERIC VALUE FROM THE COMMAND LINE.
                                ;       THE VALUE MAY BE OCTAL, HEX, OR BINARY.
                                ;
                                ;
013216                          PACK::
013216  005046                          CLR     -(SP)           ;CLEAR FOR COND CODE INDICATOR
013220                                  SAVE    R0,R1,R2,R3,R4,R5
013220  010046                          MOV     R0,-(SP)
013222  010146                          MOV     R1,-(SP)
013224  010246                          MOV     R2,-(SP)
013226  010346                          MOV     R3,-(SP)
013230  010446                          MOV     R4,-(SP)
013232  010546                          MOV     R5,-(SP)
                                ;
013234  022700  000006                  CMP     #6,R0           ;UPPER LIMIT ON OCTAL DIGITS
013240  002413                          BLT     PSECX           ;ERROR EXIT
013242  010146                          MOV     R1,-(SP)        ;SAVE STRING POINTER TEMPORARILY
013244  060016                          ADD     R0,(SP)         ;ADD CHAR COUNT
013246  010100                          MOV     R1,R0           ;GET STRING ADDR INTO R0 FOR SUBRTN
013250  004767  000000G                 JSR     PC,$COTB        ;CONVERT ASCII OCTAL
013254  005300                          DEC     R0              ;SUBRTN PUSHES R0 1 TOO FAR
013256  020026                          CMP     R0,(SP)+        ;FULL STRING CONVERTED
013260  001003                          BNE     PSECX           ;NO, ERROR
013262  010167  164524                  MOV     R1,BINWD        ;SAVE CONVERTED VALUE
013266  000403                          BR      PCLCX           ;AND RETURN
                                ;
013270  012766  177777  000014  PSECX:  MOV     #-1,12.(SP)     ;SET COND CODE INDICATOR
013276                          PCLCX:  RESTOR  R0,R1,R2,R3,R4,R5
013276  012605                          MOV     (SP)+,R5
013300  012604                          MOV     (SP)+,R4
013302  012603                          MOV     (SP)+,R3
013304  012602                          MOV     (SP)+,R2
013306  012601                          MOV     (SP)+,R1
013310  012600                          MOV     (SP)+,R0
                                ;
013312  005726                          TST     (SP)+           ;GET COND CODE INDICATOR
013314  002402                          BLT     1$              ;DO SEC
013316  000241                          CLC
013320  000401                          BR      PACKX           ;AND RETURN
013322  000261                  1$:     SEC
013324  000207                  PACKX:  RTS     PC
```

```
                                        ;
                                        ;
                                        ;       WRITE A PRINT LINE TO TT0
                                        ;
                                        ;
013372                                  CONSOL::
013372                                          SAVE    R0,R1
013372  010046                                  MOV     R0,-(SP)
013374  010146                                  MOV     R1,-(SP)
                                        ;
013376  012700  000120                          MOV     #80.,R0         ;PRINT BUFFER BYTE COUNT
013402  012701  005772'                         MOV     #PRINT+78.,R1   ;POINT PAST END OF BUFFER
013406  122741  000040          1$:     CMPB    #40.,-(R1)      ;LOOK FOR A NON-BLANK
013412  001003                          BNE     2$              ;OK, WRITE LINE
013414  005300                          DEC     R0              ;DEC CHAR COUNT
013416  001373                          BNE     1$
013420  000440                          BR      ABEND2          ;NO NON-BLANKS?
                                        ;
013422                          2$:     QIOW$S  #IO.WVB,#LUN.TT,#EFN.1,,#STAT,,<#PRINT-2,R0>,ABEND2
013422  005046                          CLR     -(SP)
013424  005046                          CLR     -(SP)
013426  005046                          CLR     -(SP)
013430  005046                          CLR     -(SP)
013432  010046                          MOV     R0,-(SP)
013434  012746  005652'                 MOV     #PRINT-2,-(SP)
013440  005046                          CLR     -(SP)
013442  012746  000004'                 MOV     #STAT,-(SP)
013446  005046                          CLR     -(SP)
013456  112716  000001                  MOVB    #EFN.1,(SP)
013454  012746  000001                  MOV     #LUN.TT,-(SP)
013460  012746  000000G                 MOV     #IO.WVB,-(SP)
013464  012746                          MOV     (PC)+,-(SP)
013466     003     014                  .BYTE   3,12.
013470  104377                          EMT     ^O<377>
013472  103002                          BCC     .+6
013474  004767  000022                  JSR     PC,ABEND2
                                        ;
                                        ;
013500  012701  005654'                 MOV     #PRINT,R1       ;POINT TO STRING
013504  112721  000040          4$:     MOVB    #40.,(R1)+      ;CLEAR LINE TO BLANKS
013510  005300                          DEC     R0              ;DEC LOOP COUNT
013512  001374                          BNE     4$
                                        ;
013514                                  RESTOR  R0,R1
013514  012601                          MOV     (SP)+,R1
013516  012600                          MOV     (SP)+,R0
013520  000207                          RTS     PC
                                        ;
013522                          ABEND2: ABRT$S  #MYSELF
013522  012746  000000'                 MOV     #MYSELF,-(SP)
013526  017646  000000                  MOV     @(SP),-(SP)
013532  062766  000002  000002          ADD     #2,2(SP)
013540  017666  000002  000002          MOV     @2(SP),2(SP)
013546  012746                          MOV     (PC)+,-(SP)
013550     123     003                  .BYTE   83.,3
013552  104377                          EMT     ^O<377>
```

```
1242·                                   ;
1243                                    ;       CONVERT·A·VALUE·FROM·BINARY·TO·PRINTABLE·FORM.
1244                                    ;       R1 =·WORD·TO BE·CONVERTED·
1245                                    ;       R5 -> PRINT·LINE·
1246                                    ;
1247                                    ;
1248 013326                            UNPK::
1249 013326                                    SAVE·    R0,R1,R2,R3,R4
     013326  010046                            MOV·     R0,-(SP)
     013330  010146                            MOV·     R1,-(SP)
     013332· 010246                            MOV·     R2,-(SP)
     013334  010346                            MOV·     R3,-(SP)
     013336  010446                            MOV·     R4,-(SP)
1250                                    ;
1251 013340  010500                            MOV·     R5,R0           ;PREPARE TO·CALL·SYSTEM·SUBRTN·
1252·013342·  012702·  000001·                  MOV·     #1,R2·          ;KEEP LEADING ZEROS
1253 013346  004767  000000G·                  JSR·     PC,$CBOMG·      ;CONVERT TO·OCTAL·ASCII·
1254 013352· 010005                            MOV·     R0,R5           ;RESTORE PRINT·LINE·POINTER
1255 013354  005205                            INC·     R5              ;AND·BUMP·IT·
1256                                    ;
1257 013356                            UNPKX:
1258 013356                                    RESTOR·  R0,R1,R2,R3,R4
     013356  012604                            MOV·     (SP)+,R4
     013360  012603                            MOV·     (SP)+,R3
     013362· 012602·                           MOV·     (SP)+,R2·
     013364  012601                            MOV·     (SP)+,R1
     013366  012600                            MOV·     (SP)+,R0
1259 013370  000207                            RTS·     PC·
```

```
1290                                    ;
1291                                    ;
1292                                    ;       WRITE TO TT0 AND PROMPT.
1293                                    ;
1294                                    ;
1295 013554  005267  164230            ENDTST:: INC     ERWORD
1296 013560  005267  164224            ENFILE:: INC     ERWORD
1297 013564  005267  164220            ENDMEM:: INC     ERWORD
1298 013570  005267  164214            STOP::   INC     ERWORD
1299 013574  005267  164210            OUT1::   INC     ERWORD
1300 013600  005267  164204            ERR23::  INC     ERWORD
1301 013604  005267  164200            ERR22::  INC     ERWORD
1302 013610  005267  164174            ERR21::  INC     ERWORD
1303 013614  005267  164170            ERR20::  INC     ERWORD
1304 013620  005267  164164            ERR19::  INC     ERWORD
1305 013624  005267  164160            ERR18::  INC     ERWORD
1306 013630  005267  164154            ERR17::  INC     ERWORD
1307 013634  005267  164150            ERR16::  INC     ERWORD
1308 013640  005267  164144            ERR15::  INC     ERWORD
1309 013644  005267  164140            ERR14::  INC     ERWORD
1310 013650  005267  164134            ERR13::  INC     ERWORD
1311 013654  005267  164130            ERR12::  INC     ERWORD
1312 013660  005267  164124            ERR11::  INC     ERWORD
1313 013664  005267  164120            ERR10::  INC     ERWORD
1314 013670  005267  164114            ERR9::   INC     ERWORD
1315 013674  005267  164110            ERR8::   INC     ERWORD
1316 013700  005267  164104            ERR7::   INC     ERWORD
1317 013704  005267  164100            ERR6::   INC     ERWORD
1318 013710  005267  164074            ERR5::   INC     ERWORD
1319 013714  005267  164070            ERR4::   INC     ERWORD
1320 013720  005267  164064            ERR3::   INC     ERWORD
1321 013724  005267  164060            ERR2::   INC     ERWORD
1322 013730  005267  164054            ERR1::   INC     ERWORD
1323                                    ;
1324 013734                            NESTOP:
1325 013734  005267  164050            LPTST::  INC     ERWORD
1326 013740  005267  164044            SELTST:: INC     ERWORD
1327 013744  005267  164040            BASEL::  INC     ERWORD
1328         000003                     NEST     =       <.-NESTOP>/4
1329                                    ;
1330                                    ;       USE THE INDEX ERWORD TO COUNT UP FROM THE BOTTOM OF
1331                                    ;       THE MESSAGE TABLE.  FIND THE END OF THE MESSAGE FIRST,
1332                                    ;       THEN THE BEGINNING.  THEN GET THE LENGTH.
1333                                    ;
1334 013750  016702  164034                    MOV     ERWORD,R2       ;LOAD LOOP COUNT
1335 013754  012701  007613                    MOV     #ASCIZ,R1       ;POINT TO END OF MESSAGE TABLE
1336 013760  105741                    1$:     TSTB    -(R1)           ;LOOK FOR END OF MESSAGE
1337 013762  001376                            BNE     1$
1338 013764  005302                            DEC     R2              ;LOOP COUNT
1339 013766  001374                            BNE     1$              ;BACK UP ANOTHER MESSAGE
1340 013770  010100                            MOV     R1,R0           ;SAVE POINTER TO END OF MESSAGE
1341 013772  105741                    2$:     TSTB    -(R1)           ;BACK UP TO BEGINNING OF MESSAGE
1342 013774  001376                            BNE     2$
1343 013776  005201                            INC     R1              ;BUMP TO FIRST CHAR OF MESSAGE
1344 014000  160100                            SUB     R1,R0           ;R0 NOW = MESSAGE LENGTH
1345                                    ;
1346 014002                                    QIOW$S  #IO.WVB,#LUN.TT,#EFN.1,,#STAT,,<R1,R0>,ABEND
```

```
        014002  005046                          CLR     -(SP)
        014004  005046                          CLR     -(SP)
        014006  005046                          CLR     -(SP)
        014010  005046                          CLR     -(SP)
        014012  010046                          MOV     R0,-(SP)
        014014  010146                          MOV     R1,-(SP)
        014016  005046                          CLR     -(SP)
        014020  012746  000004'                 MOV     #STAT,-(SP)
        014024  005046                          CLR     -(SP)
        014026  112716  000001                  MOVB    #EFN.1,(SP)
        014032  012746  000001                  MOV     #LUN.TT,-(SP)
        014036  012746  000000G                 MOV     #IO.WVB,-(SP)
        014042  012746                          MOV     (PC)+,-(SP)
        014044     003     014                  .BYTE   3,12.
        014046  104377                          EMT     ^O<377>
        014050  103002                          BCC     .+6
        014052  004767  000114                  JSR     PC,ABEND
  1347                                  ;
  1348 014056                                   CLEF$S  #EFN.1
        014056  012746  000001                  MOV     #EFN.1,-(SP)
        014062  012746                          MOV     (PC)+,-(SP)
        014064     037     002                  .BYTE   31.,2
        014066  104377                          EMT     ^O<377>
  1349 014070  105767  163710                  TSTB    STAT            ;GOOD RETURN
  1350 014074  003436                          BLE     ABEND           ;NO
  1351                                  ;
  1352                                  ;       ISSUE GCML
  1353                                  ;
  1354 014076  022767  000003  163704          CMP     #NEST,ERWORD    ;PROMPT WITH MESSAGE
  1355 014104  002427                          BLT     TTX             ;NO, JUST EXIT
  1356 014106  012700  000020'                 MOV     #GCMBUF,R0      ;POINT TO COMMAND LINE BUFFER
  1357 014112  012701  000120                  MOV     #80.,R1         ;NUMBER OF BYTES IN BUFFER
  1358 014116  112720  000040          3$:     MOVB    #40.,(R0)+      ;CLEAR COMMAND LINE
  1359 014122  005301                          DEC     R1              ;FINISHED?
  1360 014124  001374                          BNE     3$              ;NO
  1361                                  ;
  1362                                  ;       ISSUE 'GET COMMAND LINE'
  1363                                  ;       SET UP FIELDS:
  1364                                  ;       GCMBUF  - WORK AREA TO HOLD COMMAND LINE
  1365                                  ;       GCMLEN  - LENGTH OF LINE READ
  1366                                  ;       GCMPNT  - POINTER TO COMMAND LINE (SET TO POINT TO BEGINNING)
  1367                                  ;
  1368 014126                                   GCML$   #GCMBLK
        014126  012700  007614'                 MOV     #GCMBLK,R0
        014132  005060  000142                  CLR     G.PSDS(R0)
        014136  004767  000000G                 JSR     PC,.GCML1
  1369 014142  103413                          BCS     ABEND
  1370 014144  016067  000146  163770          MOV     G.CMLD(R0),GCMLEN ;SAVE LENGTH
  1371 014152  012701  000020'                 MOV     #GCMBUF,R1      ;POINT TO COMMAND BUFFER
  1372 014156  012767  000020'  163760         MOV     #GCMBUF,GCMPNT  ;INITIALIZE COMMAND BUFFER POINTER
  1373 014164  005067  163620          TTX:    CLR     ERWORD          ;CLEAR ERROR NUMBER/WINDICATOR
  1374 014170  000207                          RTS     PC              ;AND RETURN
  1375                                  ;
  1376 014172                           ABEND:  ABRT$S  #MYSELF
        014172  012746  000000'                 MOV     #MYSELF,-(SP)
        014176  017646  000000                  MOV     @(SP),-(SP)
        014202  062766  000002  000002          ADD     #2,2(SP)
```

```
014210  017666  000002  000002.          MOV.    @2(SP),2(SP)
014216  012746                           MOV.    (PC)+,-(SP)
014220     123     003                   .BYTE.  83.,3
014222. 104377                           EMT.    +0<377>
1377                                ;
1378          010516'                    .END.   START.
```

```
ABEND   014172R
ABEND2  013522R
AHIGH   000206RG
ALOW    000210RG
ALUCKE= 040000
ALUOE = 004000
APLACE  000016RG
ASCIZ   007613R
AST     012766R
ASTFLG= 000020 G
ASTWRD  000146R
A01   = 010000
BASE    000014RG
BASEL   013744RG
BHIGH   000212RG
BINWD   000012RG
BITVAL= 000000
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BLOW    000214RG
BTEND   005450RG
BTOVER  005452RG
BTRANS  005250RG
BUFCX   010770R
BUFCX2  011206R
BUFFSA  010626RG
BUFSET  011014RG
BUFS2   011004RG
BUFXX   011002R
BUFXX2  011220R
BYTE0 = 000000
BYTE1 = 000001
BYTE10= 000012
BYTE11= 000013
BYTE12= 000014
BYTE13= 000015
BYTE14= 000016
BYTE15= 000017
BYTE16= 000020
BYTE17= 000021
BYTE18= 000022
BYTE19= 000023
BYTE2 = 000002
BYTE20= 000024
BYTE21= 000025
BYTE22= 000026
BYTE23= 000027
BYTE24= 000030
BYTE25= 000031
BYTE26= 000032
BYTE27= 000033
BYTE28= 000034
BYTE29= 000035
BYTE3 = 000003
BYTE30= 000036
BYTE31= 000037
BYTE32= 000040
BYTE33= 000041
BYTE34= 000042
BYTE35= 000043
BYTE36= 000044
BYTE37= 000045
BYTE38= 000046
BYTE39= 000047
BYTE4 = 000004
BYTE40= 000050
BYTE41= 000051
BYTE42= 000052
BYTE43= 000053
BYTE44= 000054
BYTE45= 000055
BYTE46= 000056
BYTE47= 000057
BYTE48= 000060
BYTE49= 000061
BYTE5 = 000005
BYTE50= 000062
BYTE51= 000063
BYTE52= 000064
BYTE53= 000065
BYTE54= 000066
BYTE55= 000067
BYTE56= 000070
BYTE57= 000071
BYTE58= 000072
BYTE59= 000073
BYTE6 = 000006
BYTE60= 000074
BYTE61= 000075
BYTE62= 000076
BYTE63= 000077
BYTE64= 000100
BYTE65= 000101
BYTE66= 000102
BYTE67= 000103
BYTE68= 000104
BYTE69= 000105
BYTE7 = 000007
BYTE70= 000106
BYTE71= 000107
BYTE72= 000110
BYTE73= 000111
BYTE74= 000112
BYTE75= 000113
BYTE76= 000114
BYTE77= 000115
BYTE78= 000116
BYTE79= 000117
BYTE8 = 000010
BYTE80= 000120
BYTE81= 000121
BYTE82= 000122
BYTE83= 000123
BYTE84= 000124
BYTE85= 000125
BYTE86= 000126
BYTE87= 000127
BYTE88= 000130
BYTE89= 000131
BYTE9 = 000011
BYTE90= 000132
BYTE91= 000133
BYTE92= 000134
BYTE93= 000135
BYTE94= 000136
BYTE95= 000137
BYTE96= 000140
BYTE97= 000141
BYTE98= 000142
BYTE99= 000143
BYTVAL= 000144
CBKALL= 001000
CBKCLK= 000400
CHIGH   000216RG
CHLEN   000176RG
CH1   = ****** GX
CLOW    000220RG
CMILUN= 000002 G
CNOBRE= 100000
CNTLX   011400R
CNUM  = 000015 G
COMXX   010522RG
CONSOL  013372RG
COUNT   000166RG
CPCCEN= 010000
CPREAD= 040000
CPWRTE= 020000
CSADRD= 000004
CSEQCI= 100000
CSOE  = 000040
CSWRTE= 000100
CTBL    000236RG
DATALN  000170RG
DATA1   000200RG
DATA2   000202RG
DATA3   000204RG
DA1   = ****** GX
DBLD    012074RG
DBLDCX  012504R
DBLDSX  012474R
DBLDXX  012506R
DBLDZ   012102R
DBR.RD= 000001
DB$CPP= 001457
DB$SPT= 000026
DB$TPC= 000023
DF1   = ****** GX
DISPGS= 100000
DMAAWR= 000005
DMARRD= 000003
DMARWR= 000004
DPLUS   000172RG
DSAVE   002322RG
DSEND   005046RG
DTBL    000322RG
EFN.1 = 000001 G
ENBR  = 010000
ENDMEM  013564RG
ENDTST  013554RG
ENFILE  013560RG
EN1   = ****** GX
ERR1    013730RG
ERR10   013664RG
ERR11   013660RG
ERR12   013654RG
ERR13   013650RG
ERR14   013644RG
ERR15   013640RG
ERR16   013634RG
ERR17   013630RG
ERR18   013624RG
ERR19   013620RG
ERR2    013724RG
ERR20   013614RG
ERR21   013610RG
ERR22   013604RG
ERR23   013600RG
ERR3    013720RG
ERR4    013714RG
ERR5    013710RG
ERR6    013704RG
ERR7    013700RG
ERR8    013674RG
ERR9    013670RG
ERWORD  000010R
EX1     010620R
FD.CCL= ****** GX
FD.REC= ****** GX
FD.TTY= ****** GX
FIND    013100RG
FNIN1   013022R
FNMTCH  013056R
FNOUT1  013014R
FNOUT2  013034R
FSECX   013210R
F.ACTL= 000076
F.ALOC= 000040
F.BBFS= 000062
F.BDB = 000070
F.BGBC= 000057
F.BKDN= 000026
F.BKDS= 000020
F.BKEF= 000050
F.BKP1= 000051
F.BKST= 000024
F.BKVB= 000064
F.CHR = 000075
F.CNTG= 000034
F.DFNB= 000046
F.DSPT= 000044
F.DVNM= 000134
F.EFBK= 000010
F.EFN = 000050
F.EOBB= 000032
F.ERR = 000052
F.FACC= 000043
F.FFBY= 000014
F.FNAM= 000110
F.FNB = 000102
F.FTYP= 000116
F.FVER= 000120
F.HIBK= 000004
F.LUN = 000042
F.MBCT= 000054
F.MBC1= 000055
F.MBFG= 000056
F.NRBD= 000024
F.NREC= 000030
F.OVBS= 000030
F.RACC= 000016
F.RATT= 000001
F.RCNM= 000034
F.RCTL= 000017
F.RSIZ= 000002
F.RTYP= 000000
F.SEQN= 000100
F.SPDV= 000072
F.SPUN= 000074
F.STBK= 000036
F.UNIT= 000136
F.URBD= 000020
F.VBN = 000064
F.VBSZ= 000060
GCMBLK  007614R
GCMBUF  000020RG
GCMLEN  000142RG
GCMPNT  000144RG
GCONLY  012510RG
GE.BIF= 177775
GE.CLO= 000004
GE.COM= 000001
```

```
GE.CON= 000020          MYSELF  000000R          Q$CP2 = 000260          RTNPT   000150RG        T$CD  = 000100
GE.EOF= 177766          N       = 000144         Q$CSC = 010000          RU1     = ****** GX     T$CLK = 002000
GE.IND= 000002          NEST    = 000003         Q$CSEL= 000360          SCAN    = 013006RG      T$DISK= 000200
GE.IOR= 177777          NESTOP  013734R          Q$CSET= 000002          SELTST  013740RG        T$DRD = 000004
GE.LC = 000010          NEXTD   012456R          Q$CSP = 020000          SEQ.CI= 000010          T$EMEM= 010000
GE.MDE= 177774          NOCRLF  012226R          Q$DMA = 000001          SET     011022R         T$FSAA= 000000
GE.OPR= 177776          N.DID = 000024           Q$ENBK= 040000          START   010516R         T$FSAB= 000004
GE.RBG= 177730          N.DVNM= 000032           Q$ENOP= 020000          STAT    000004RG        T$FSAC= 000014
GE.SIZ= 000040          N.FID = 000000           Q$FAL = 004000          STOP    013570RG        T$FSB2= 000010
G.CMLD= 000146          N.FNAM= 000006           Q$FC  = 000045          ST1     = ****** GX     T$IB  = 000026
G.DPRM= 000160          N.FTYP= 000014           Q$FO  = 000044          S$CLR = 000000          T$IBAR= 000024
G.ERR = 000140          N.FVER= 000016           Q$FP  = 000046          S$LA  = 000001          T$IBE = 020000
G.ISIZ= 000020          N.NEXT= 000022           Q$HBF = 000002          S$QB  = 000005          T$IBF = 040000
G.LPDL= 000060          N.STAT= 000020           Q$ICP = 000006          S$QR  = 000006          T$ICD = 000040
G.MODE= 000141          N.UNIT= 000034           Q$IHB = 000003          S$QX  = 000004          T$MODE= 004000
G.PSDS= 000142          OFFSET  012006RG         Q$IHRL= 000002          S$SR  = 000007          T$OB  = 000036
G.SIZE= 000224          OFF6    000174RG         Q$IMRP= 000007          S$S1  = 000010          T$OBE = 004000
HANG    012622RG        OHIGH   000226RG         Q$LBD = 001000          S$S2  = 000014          T$OBF = 010000
IHIGH   000222RG        OLOW    000230RG         Q$LBDP= 001001          S.BFHD= 000020          T$OBRA= 000034
ILOW    000224RG        ONCE    = 000010 G       Q$LBP = 000001          S.FATT= 000016          T$OBWA= 000032
INCVAL  000160RG        OUT     = 002000 G       Q$LDCD= 000003          S.FDB = 000140          T$OUTA= 100000
IO.ATA= ****** G        OUT1    013574RG         Q$LDMD= 000004          S.FNAM= 000006          T$RBDO= 000200
IO.DET= ****** G        PACK    013216RG         Q$LDPP= 002000          S.FNB = 000036          T$RNB = 000040
IO.RVB= ****** G        PACKX   013324R          Q$LHP = 010000          S.FNBW= 000017          T$RSET= 040000
IO.WVB= ****** G        PAR$$$= 000027           Q$MNC = 140000          S.FNTY= 000004          T$SC  = 000022
KILL    012706RG        PCLCX   013276R          Q$MR  = 000852          S.FTYP= 000002          T$SCLK= 020000
LDCNTL  011222RG        PCNTLX  011706R          Q$MRP = 000040          S.NFEN= 000020          T$SEG1= 000000
LDI     = ****** GX     PDATA   011402RG         Q$MRP2= 000240          TD$CTR= 176370          T$SEG2= 000001
LII     = ****** GX     PDCCX   011606R          Q$MSC = 040000          TD$CTW= 176360          T$SEG3= 000002
LOC.EN= 000100          PDCX    011600R          Q$MSET= 000004          TD$INL= 004000          T$SO  = 000001
LOC.WA= 040000          PDDX    011612R          Q$MSP = 100000          TD$MEM= 000270          T$UBUS= 100000
LOC.WB= 100000          PDVX    011572R          Q$NCLK= 176000          TD$OAR= 176344          T$1CLK= 000400
LOOP  = 000004 G        PLB     = 000010         Q$PP  = 000100          TD$OTR= 176346          T$8BEN= 000020
LOOPR   012566RG        PLC     = 000020         Q$PPSW= 000320          TD$QRD= 000274          UBD.IN= 000020
LOOPX   012704R         PLD     = 000030         Q$PP2 = 000300          TD$SW = 176376          UNPK    013326RG
LPTST   013734RG        PLRWR = 000200           Q$QHLT= 000013          TD$TAR= 176372          UNPKX   013356R
LUN.TT= 000001 G        PLR.EN= 000200           Q$QL  = 000043          TD$TAW= 176362          UPLIM   000164RG
MAREN1= 000001          PRCNTL  011614RG         Q$QLA = 000053          TD$TDR= 176374          WORD0 = 000000
MAREN2= 004000          PRDATA  011710RG         Q$QLB = 000054          TD$TDW= 176364          WORD1 = 000002
MARLOD= 010000          PRDX    012004R          Q$QLR = 000001          THIGH   000232RG        WORD10= 000024
MAROUT= 000002          PRINT   005654RG         Q$QW  = 000042          TLOW    000234RG        WORD11= 000026
MAR.LO= 002000          PRI     = ****** GX      Q$RDCD= 000005          TRFDB   010320RG        WORD12= 000030
MAR.OU= 000040          PSECX   013270R          Q$RDMD= 000006          TRLUN = 000004 G        WORD13= 000032
MBKALL= 001000          QN      = 001000 G       Q$REBK= 001000          TRNMB   010450R         WORD14= 000034
MBKCLK= 000400          QP      = 000400 G       Q$RNC = 006000          TR1     = ****** GX     WORD15= 000036
MEND    000156RG        QR$CR1= 176420           Q$RSC = 004000          TR6TBL  005050RG        WORD16= 000040
MMADRD= 000100          QR$CR2= 176422           Q$RSET= 000010          TTX     014164R         WORD17= 000042
MMLEFT= 000002          QR$LBR= 176424           Q$SM  = 100000          TXTFDB  010122RG        WORD18= 000044
MMOE  = 000004          Q$ATTN= 000100           Q$SP  = 000120          TXTLUN= 000003 G        WORD19= 000046
MMWRTE= 000010          Q$BCL = 000001           Q$SP2 = 000340          TXTNMB  010262R         WORD2 = 000004
MNOBRE= 100000          Q$CCCP= 000040           RE1     = ****** GX     T$AD  = 000020          WORD20= 000050
MOVTR   012252R         Q$CHB = 000400           RGQ.EN= 000200          T$BA  = 000002          WORD21= 000052
MREN1 = 000001          Q$CHRL= 000200           RGQ.VA= 020000          T$BD  = 000010          WORD22= 000054
MREN2 = 020000          Q$CLR = 000040           RIDE  = 000100 G        T$BSO = 100000          WORD23= 000056
MSTRT   000152RG        Q$CNC = 030000           RP      = 000200 G      T$BT  = 000020          WORD24= 000060
MSTR2   000154RG        Q$CP  = 000060           RSPCNT  000162RG        T$BTAR= 000030          WORD25= 000062
MSYN  = 000040          Q$CPCC= 000010           RS1     010606R         T$BTD = 002000          WORD26= 000064
```

```
WORD27=  000066          WORD45=  000132          WORD63=  000176          WORD81=  000242          WORD99=  000306
WORD28=  000070          WORD46=  000134          WORD64=  000200          WORD82=  000244          WRDVAL=  000310
WORD29=  000072          WORD47=  000136          WORD65=  000202          WORD83=  000246          WWORDS=  000200RG
WORD3 =  000006          WORD48=  000140          WORD66=  000204          WORD84=  000250          XTREAD=  001000
WORD30=  000074          WORD49=  000142          WORD67=  000206          WORD85=  000252          XTWRTE=  000400
WORD31=  000076          WORD5 =  000012          WORD68=  000210          WORD86=  000254          $CBOMG=  ****** GX
WORD32=  000100          WORD50=  000144          WORD69=  000212          WORD87=  000256          $COTB =  ****** GX
WORD33=  000102          WORD51=  000146          WORD7 =  000016          WORD88=  000260          $DIV  =  ****** GX
WORD34=  000104          WORD52=  000150          WORD70=  000214          WORD89=  000262          $MUL  =  ****** GX
WORD35=  000106          WORD53=  000152          WORD71=  000216          WORD9 =  000022          $$$   =  007774R
WORD36=  000110          WORD54=  000154          WORD72=  000220          WORD90=  000264          $$$ARG=  000002
WORD37=  000112          WORD55=  000156          WORD73=  000222          WORD91=  000266          $$$T1 =  000067
WORD38=  000114          WORD56=  000160          WORD74=  000224          WORD92=  000270          $$$T2 =  000027
WORD39=  000116          WORD57=  000162          WORD75=  000226          WORD93=  000272          .FSRCB=  ****** G
WORD4 =  000010          WORD58=  000164          WORD76=  000230          WORD94=  000274          .GCML1=  ****** G
WORD40=  000120          WORD59=  000166          WORD77=  000232          WORD95=  000276          ...PC1=  010320R
WORD41=  000122          WORD6 =  000014          WORD78=  000234          WORD96=  000300          ...PC2=  010474R
WORD42=  000124          WORD60=  000170          WORD79=  000236          WORD97=  000302          ...PC3=  010320R
WORD43=  000126          WORD61=  000172          WORD8 =  000020          WORD98=  000304          ...TPC=  000020
WORD44=  000130          WORD62=  000174          WORD80=  000240

. ABS.  000000     000
        014224     001
$$FSR1  002040     002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  8170 WORDS  ( 32 PAGES)
DYNAMIC MEMORY:  9140 WORDS  ( 35 PAGES)
ELAPSED TIME:  00:01:42
!MAIN,TMAIN/-SP=[20,1]IM,[20,1]TMAIN
```

```
                                        .TITLE  TREG
000000                                  .PSECT  TREG
                                        .LIST   MEB
                                ;
                                ;
                                ;       TERM DETECTOR 'MANUAL' DEBUGGING AIDS
                                ;       REGISTER EXERCISES
                                ;
                                ;       COMMANDS:
                                ;       ST      STORE INTO A TD REGISTER
                                ;       RE      READ FROM A TD REGISTER
                                ;
                                ;       ONCE A COMMAND HAS BEEN EXECUTED (OR AN ERROR ENCOUNTERED)
                                ;       THIS MODULE RETURNS CONTROL TO THE MODULE TMAIN AT LOCATION
                                ;       'COMXX'.
                                ;
                                ;
                                ;       STORE
                                ;       TABLE OF REGISTER MNEMONICS AND ASSOCIATED ROUTINE
                                ;       ADDRESSES.
                                ;
000000                          ST1TBL::
000000  101     101                     .ASCII  /AA/            ;FSA A MEMORY ADDRESS REGISTER
000002  000224'                         .WORD   STA
000004  102     101                     .ASCII  /BA/            ;FSA B MEMORY ADDRESS REGISTER
000006  000234'                         .WORD   STB
000010  103     101                     .ASCII  /CA/            ;FSA C MEMORY ADDRESS REGISTER
000012  000244'                         .WORD   STC
000014  111     102                     .ASCII  /IB/            ;INPUT BUFFER MEMORY ADDRESS REGISTER
000016  000276'                         .WORD   STI
000020  117     102                     .ASCII  /OB/            ;OUTPUT BUFFER MEMORY ADDRESS REGISTER
000022  000322'                         .WORD   STO
000024  102     124                     .ASCII  /BT/            ;BYTE TRANSLATOR MEMORY ADDRESS REGISTER
000026  000346'                         .WORD   STBT
000030  123     103                     .ASCII  /SC/            ;SECTOR COUNTER
000032  000372'                         .WORD   STSC
000034  103     122                     .ASCII  /CR/            ;CONTROL REGISTER
000036  000416'                         .WORD   STCR
        000010          ST1LN   ==      <.-ST1TBL>/4
                                ;
                                ;       READ
                                ;
000040                          RE1TBL::
000040  101     101                     .ASCII  /AA/            ;FSA A MEMORY ADDRESS REGISTER
000042  000526'                         .WORD   REAA
000044  102     101                     .ASCII  /BA/            ;FSA B MEMORY ADDRESS REGISTER
000046  000542'                         .WORD   REBA
000050  103     101                     .ASCII  /CA/            ;FSA C MEMORY ADDRESS REGISTER
000052  000556'                         .WORD   RECA
000054  111     102                     .ASCII  /IB/            ;INPUT BUFFER MEMORY ADDRESS REGISTER
000056  000572'                         .WORD   REIA
000060  117     102                     .ASCII  /OB/            ;OUTPUT BUFFER MEMORY ADDRESS REGISTER
000062  000614'                         .WORD   REOA
000064  102     124                     .ASCII  /BT/            ;BYTE TRANSLATOR MEMORY ADDRESS REGISTER
000066  000636'                         .WORD   RETA
000070  123     103                     .ASCII  /SC/            ;SECTOR COUNTER
000072  000660'                         .WORD   RESC
```

```
                                .ASCII  /SW/            ;STATUS WORD
000076  000710'                 .WORD   RESW
000100      103     122         .ASCII  /CR/            ;CONTROL REGISTER
000102  000702'                 .WORD   RECR
000104      115     123         .ASCII  /MS/            ;MISC SIGNALS
000106  000716'                 .WORD   REMS
000110      121     122         .ASCII  /OR/            ;TO OUTPUT BUFFER (TO QR)
000112  000740'                 .WORD   REOR
        000013          RE1LN   ==      <.-RE1TBL>/4
                        ;
                        ;
                        ;       STORE
                        ;       PERFORM SECOND LEVEL PARSING
                        ;       EG. IN THE COMMAND:
                        ;       >ST 0 CR
                        ;       PARSE THE '0'
                        ;
000114                  ST1::
000114  004767  000000G         JSR     PC,FIND         ;LOCATE OPERAND IN COMMAND LINE
000120  103003                  BCC     1$              ;OK, CONTINUE
000122  004767  000000G         JSR     PC,ERR3         ;'MISSING OPERAND'
000126  000543                  BR      ST1X            ;EXIT
000130  004767  000000G 1$:     JSR     PC,PACK         ;CONVERT COMMAND LINE VALUE TO BINARY
000134  103003                  BCC     2$              ;CONVERSION SUCCESSFUL
000136  004767  000000G         JSR     PC,ERR4         ;'INVALID NUMERIC VALUE'
000142  000535                  BR      ST1X
                        ;
                        ;       CONTINUE PARSING
                        ;       FIND THE REGISTER MNEMONIC IN THE COMMAND LINE
                        ;
000144  004767  000000G 2$:     JSR     PC,FIND         ;LOCATE REG MNEMONIC
000150  103003                  BCC     3$              ;OK, CONTINUE
000152  004767  000000G         JSR     PC,ERR3         ;'MISSING OPERAND'
000156  000527                  BR      ST1X
                        ;
                        ;       MATCH THE MNEMONIC FROM THE COMMAND LINE AGAINST
                        ;       THE TABLE OF VALID MNEMONICS.
                        ;
000160  012700  000010  3$:     MOV     #ST1LN,R0       ;NUMBER OF TABLE ENTRIES
000164  012702  000000'         MOV     #ST1TBL,R2      ;POINT TO TABLE
000170  004767  000000G         JSR     PC,SCAN         ;MATCH AGAINST COMMAND LINE
000174  103003                  BCC     4$              ;MATCH WAS MADE
000176  004767  000000G         JSR     PC,ERR5         ;'INVALID REGISTER MNEMONIC'
000202  000515                  BR      ST1X
                        ;
                        ;       SAVE THE POINTER TO THE ROUTINE ASSOCIATED WITH THE
                        ;       REGISTER, R1 -> ROUTINE ADDRESS.
                        ;       CALL ROUTINE TO SCAN THE COMMAND LINE FOR A LOOP
                        ;       INDICATOR, EG:
                        ;       >ST 0 CR L
                        ;       LOOP FLAG WILL BE SET IF INDICATOR IS PRESENT.
                        ;       JUMP TO ROUTINE TO LOAD REGISTER.
                        ;
000204  010167  000000G 4$:     MOV     R1,RTNPT        ;SAVE POINTER
000210  004767  000000G         JSR     PC,LOOPR        ;LOOP ON TEST
000214  016701  000000G STIN:   MOV     RTNPT,R1        ;POINT TO ROUTINE
000220  000171  000000          JMP     @(R1)           ;JUMP TO ROUTINE
```

```
                                        ;
                                        ;
                                        ;       STORE INTO THE MEMORY ADDRESS REGISTERS FOR THE FSA'S
                                        ;
000224  012767  000000  000000G STA::    MOV     #T$FSAA,APLACE  ;SELECT FSA A
000232  000407                          BR      STOR            ;MOVE SELECTION TO TD
000234  012767  000004  000000G STB::    MOV     #T$FSAB,APLACE  ;SELECT FSA B
000242  000403                          BR      STOR
000244  012767  000014  000000G STC::    MOV     #T$FSAC,APLACE  ;SELECT FSA C
                                        ;
000252  052767  020000  000000G STOR:    BIS     #T$SCLK,APLACE  ;START CLOCK
000260  016767  000000G 176360          MOV     APLACE,TD$CTW   ;MOVE TO CONTROL REG
000266  016767  000000G 176362          MOV     BINWD,TD$TAW    ;MOVE VALUE FROM COMMAND LINE
000274  000454                          BR      STLP            ;CHECK LOOP FLAG
                                        ;
                                        ;       STORE INTO INPUT BUFFER MAR
                                        ;
000276  012767  000040  176360 STI::    MOV     #T$ICD,TD$CTW   ;ENABLE INTERFACE AND CONTROL
000304  012767  000024  176362          MOV     #T$IBAR,TD$TAW
000312  016767  000000G 176364          MOV     BINWD,TD$TDW
000320  000442                          BR      STLP            ;TEST LOOP FLAG
                                        ;
                                        ;       STORE INTO THE OUTPUT BUFFER MAR
                                        ;
000322  012767  000040  176360 STO::    MOV     #T$ICD,TD$CTW   ;ENABLE INTERFACE AND CNTL
000330  012767  000032  176362          MOV     #T$OBWA,TD$TAW  ;SELECT OUTPUT BUFFER
000336  016767  000000G 176364          MOV     BINWD,TD$TDW    ;MOVE WORD TO TRANSFER REG
000344  000430                          BR      STLP
                                        ;
                                        ;       STORE INTO THE BYTE TRANSLATOR MAR
                                        ;
000346  012767  000040  176360 STBT::   MOV     #T$ICD,TD$CTW   ;ENABLE INTERFACE AND CONTROL
000354  012767  000030  176362          MOV     #T$BTAR,TD$TAW  ;SELECT BYTE TRANSLATOR
000362  016767  000000G 176364          MOV     BINWD,TD$TDW    ;TRANSEER DATA WORD
000370  000416                          BR      STLP
                                        ;
                                        ;       STORE INTO THE SECTOR COUNTER
                                        ;
000372  012767  000040  176360 STSC::   MOV     #T$ICD,TD$CTW   ;ENABLE INTERFACE AND CONTROL
000400  012767  000022  176362          MOV     #T$SC,TD$TAW    ;SELECT SECTOR COUNTER
000406  016767  000000G 176364          MOV     BINWD,TD$TDW    ;TRANSFER DATA WORD
000414  000404                          BR      STLP
                                        ;
                                        ;       STORE INTO THE CONTROL REGISTER
                                        ;
000416  016767  000000G 176360 STCR::   MOV     BINWD,TD$CTW    ;MOVE TO CONTROL REG
000424  000400                          BR      STLP
                                        ;
                                        ;
000426  032767  000000G 000000G STLP:    BIT     #LOOP,BASE      ;LOOP FLAG ON
000434  001267                          BNE     STIN            ;YES, REPEAT
                                        ;
000436                          ST1X:
000436  004767  000000G                 JSR     PC,KILL         ;KILL AST (IF THERE WAS ONE)
000442  000167  000000G                 JMP     COMXX
```

```
                                        ;
                                        ;
                                        ;       READ·
                                        ;       PERFORM·SECOND·LEVEL·PARSING·
                                        ;       EG. IN·THE·COMMAND:
                                        ;       >RE·CR·
                                        ;       PARSE·THE·'CR'
                                        ;
                                        ;
000446                                  RE1::
000446  004767  000000G·                        JSR·    PC,FIND·        ;LOCATE·THE·REGISTER·MNEMONIC·
000452· 103003                          BCC·    1$                      ;FOUND·ONE·
000454  004767  000000G·                        JSR·    PC,ERR3         ;'MISSING·OPERAND'
000460  000574                          BR      RE1X
                                        ;
                                        ;       MATCH·THE·REGISTER·MNEMONIC·FROM THE·COMMAND·LINE·
                                        ;       AGAINST·A·TABLE·OF·VALID·MNEMONICS.
                                        ;
000462· 012700  000013          1$:     MOV·    #RE1LN,R0       ;NUMBER·OF·TABLE·ENTRIES·
000466  012702· 000040'                 MOV·    #RE1TBL,R2·     ;POINT·TO·TABLE·
000472· 004767  000000G·                JSR·    PC,SCAN·        ;MATCH·AGAINST·COMMAND·LINE·
000476  103003                          BCC·    2$              ;MATCH·WAS·MADE·
000500  004767  000000G·                JSR·    PC,ERR5         ;'INVALID·REGISTER MNEMONIC'
000504  000562·                         BR      RE1X
                                        ;
                                        ;       SAVE·THE·POINTER·TO·THE·ROUTINE·ASSOCIATED·WITH·THE·
                                        ;       REGISTER. R1 -> ROUTINE·ADDRESS.
                                        ;       CALL·ROUTINE TO·SCAN·THE·COMMAND LINE·FOR·A·LOOP·
                                        ;       INDICATOR. EG:
                                        ;       >RE·CR·L·
                                        ;       LOOP·FLAG·WILL·BE·SET·IF·INDICATOR·IS·PRESENT.
                                        ;       JUMP·TO·ROUTINE·TO·READ·REGISTER.
                                        ;
000506  010167  000000G·          2$:    MOV·    R1,RTNPT        ;SAVE·ROUTINE·POINTER·
000512· 004767  000000G·                JSR·    PC,LOOPR·       ;LOOP·ON TEST?·
000516  016701  000000G·          REIN:  MOV·    RTNPT,R1        ;POINT·TO·ROUTINE·
000522· 000171  000000                  JMP·    @(R1)           ;GO·THERE·
                                        ;
                                        ;
                                        ;       READ·FSA·A·MEMORY·ADDRESS·REGISTER·
                                        ;
000526                                  REAA::
000526  012767  000000· 176360                  MOV·    #T$FSAA,TD$CTW  ;SELECT·FSA·A·
000534  016701  176372                          MOV·    TD$TAR,R1       ;GET·REG CONTENTS·
000540  000523                          BR      REPUT·          ;PRINT·
                                        ;
                                        ;       READ·FSA·B·MEMORY·ADDRESS·REGISTER·
                                        ;
000542·                                 REBA::
000542· 012767  000004· 176360                  MOV·    #T$FSAB,TD$CTW  ;SELECT·FSA·B·
000550  016701  176372                          MOV·    TD$TAR,R1       ;GET·REG CONTENTS·
000554· 000515                          BR      REPUT·          ;AND·PRINT·
                                        ;
                                        ;       READ·FSA·C·MEMORY·ADDRESS·REGISTER·
                                        ;
000556                                  RECA::
000556  012767  000014  176360                  MOV·    #T$FSAC,TD$CTW  ;SELECT·FSA·C·
```

```
000564  016701  176372          MOV     TD$TAR,R1       ;GET REG CONTENTS
000570  000507                  BR      REPUT           ;PRINT
                        ;
                        ;       READ INPUT BUFFER MEMORY ADDRESS REGISTER
                        ;
000572                  REIA::
000572  012767  000040  176360  MOV     #T$ICD,TD$CTW   ;SELECT INTERFACE AND CONTROL
000600  012767  000024  176362  MOV     #T$IBAR,TD$TAW  ;SELECT INPUT BUF REG
000606  016701  176374          MOV     TD$TDR,R1       ;GET REG CONTENTS
000612  000476                  BR      REPUT           ;PRINT
                        ;
                        ;       READ OUTPUT BUFFER MEMORY ADDRESS REGISTER
                        ;
000614                  REOA::
000614  012767  000040  176360  MOV     #T$ICD,TD$CTW   ;SELECT INTERFACE AND CONTROL
000622  012767  000034  176362  MOV     #T$OBRA,TD$TAW  ;SELECT OUTPUT BUFFER MAR
000630  016701  176374          MOV     TD$TDR,R1       ;GET REG CONTENTS
000634  000465                  BR      REPUT           ;PRINT
                        ;
                        ;       READ BYTE TRANSLATOR MEMORY ADDRESS REGISTER
                        ;
000636                  RETA::
000636  012767  000040  176360  MOV     #T$ICD,TD$CTW   ;SELECT INTERFACE AND CONTROL
000644  012767  000030  176362  MOV     #T$BTAR,TD$TAW  ;SELECT BYTE TRANSLATOR
000652  016701  176374          MOV     TD$TDR,R1       ;GET BYTE TRANS MAR
000656  000454                  BR      REPUT           ;PRINT
                        ;
                        ;       READ SECTOR COUNTER
                        ;
000660                  RESC::
000660  012767  000040  176360  MOV     #T$ICD,TD$CTW   ;SELECT INTERFACE AND CONTROL
000666  012767  000022  176362  MOV     #T$SC,TD$TAW    ;SELECT SECTOR COUNTER
000674  016701  176374          MOV     TD$TDR,R1       ;READ SECTOR COUNTER
000700  000443                  BR      REPUT           ;AND PRINT
                        ;
                        ;       READ CONTROL REGISTER
                        ;
000702                  RECR::
000702  016701  176370          MOV     TD$CTR,R1       ;LOAD CONTROL REG
000706  000440                  BR      REPUT           ;PRINT IT
                        ;
                        ;       READ STATUR REGISTER
                        ;
000710  016701  176376  RESW::  MOV     TD$SW,R1        ;LOAD STATUS REG
000714  000435                  BR      REPUT           ;PRINT IT
                        ;
                        ;       READ MISC CONTROL SIGNALS
                        ;
000716                  REMS::
000716  012767  000040  176360  MOV     #T$ICD,TD$CTW   ;SELECT INTERFACE AND CONTROL
000724  012767  000030  176362  MOV     #T$BTAR,TD$TAW  ;LOAD MISC REG
000732  016701  176374          MOV     TD$TDR,R1       ;READ MISC REG
000736  000424                  BR      REPUT
                        ;
                        ;       PRINT TD OUTPUT TO QR
                        ;
000740                  REOR::
```

```
  285 000740  016701  176344                    MOV·    TD$OAR,R1       ;READ·OUTPUT·AVAILABLE·REG·
  286 000744  032701  100000                    BIT·    #T$OUTA,R1      ;IS·OUTPUT·AVAILABLE·
  287 000750  001006                            BNE·    1$              ;YES, PRINT·
  288 000752· 012767  046505  000000G·          MOV·    #"EM,PRINT·     ;EMPTY·
  289 000760  004767  000000G·                  JSR·    PC,CONSOL·      ;WRITE·MESSAGE·
  290 000764  000426                            BR      RELP            ;TEST·LOOP·
  291 000766  012705  000000G·          1$:     MOV·    #PRINT,R5       ;POINT·TO·PRINT·LINE·
  292·000772· 016701  176346                    MOV·    TD$OTR,R1       ;READ·OUTPUT·
  293 000776  004767  000000G·                  JSR·    PC,UNPK·        ;CONVERT
  294 001002· 004767  000000G·                  JSR·    PC,CONSOL·      ;PRINT·IT·
  295 001006  000415                            BR      RELP            ;TEST·LOOP·
  296                                 ;
  297                                 ;       PRINT·REGISTER·CONTENTS·
  298                                 ;
  299 001010                          REPUT:
  300 001010  032767  000000G·000000G·          BIT·    #ONCE,BASE·     ;CONTENTS·ALREADY·PRINTED·ONCE·
  301 001016  001011                            BNE·    RELP            ;YES, SKIP·SUBSEQUENT·PRINTINGS·
  302·001020  052767  000000G·000000G·          BIS·    #ONCE,BASE·     ;SET·FLAG·FOR·LINE·PRINTED·
  303 001026  012705  000000G·                  MOV·    #PRINT,R5       ;SET·UP·POINTER·TO·PRINT·LINE·
  304 001032· 004767  000000G·                  JSR·    PC,UNPK·        ;CONVERT
  305 001036  004767  000000G·                  JSR·    PC,CONSOL·      ;WRITE·TO·TT0
  306                                 ;
  307 001042· 032767  000000G·000000G·RELP:     BIT·    #LOOP,BASE·     ;LOOP·FLAG·ON·
  308 001050  001222·                           BNE·    REIN            ;YES, REPEAT·
  309                                 ;
  310 001052·                         REIX:
  311 001052· 042767  000000G·000000G·          BIC·    #ONCE,BASE·     ;CLEAR·LOCAL·FLAG·
  312·001060  004767  000000G·                  JSR·    PC,KILL·        ;KILL·AST·(IF·THERE·WAS·ONE)
  313 001064  000167  000000G·                  JMP·    COMXX·
  314                                 ;
  315         000001                            .END·
```

```
ALUCKE= 040000
ALUOE = 004000
APLACE= ****** GX
A01   = 010000
BASE  = ****** GX
BINWD = ****** GX
BITVAL= 000000
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BYTE0 = 000000
BYTE1 = 000001
BYTE10= 000012
BYTE11= 000013
BYTE12= 000014
BYTE13= 000015
BYTE14= 000016
BYTE15= 000017
BYTE16= 000020
BYTE17= 000021
BYTE18= 000022
BYTE19= 000023
BYTE2 = 000002
BYTE20= 000024
BYTE21= 000025
BYTE22= 000026
BYTE23= 000027
BYTE24= 000030
BYTE25= 000031
BYTE26= 000032
BYTE27= 000033
BYTE28= 000034
BYTE29= 000035
BYTE3 = 000003
BYTE30= 000036
BYTE31= 000037
BYTE32= 000040
BYTE33= 000041
BYTE34= 000042
BYTE35= 000043
BYTE36= 000044
BYTE37= 000045
BYTE38= 000046
BYTE39= 000047
BYTE4 = 000004
BYTE40= 000050
BYTE41= 000051
BYTE42= 000052
BYTE43= 000053
BYTE44= 000054
BYTE45= 000055
BYTE46= 000056
BYTE47= 000057
BYTE48= 000060
BYTE49= 000061
BYTE5 = 000005
BYTE50= 000062
BYTE51= 000063
BYTE52= 000064
BYTE53= 000065
BYTE54= 000066
BYTE55= 000067
BYTE56= 000070
BYTE57= 000071
BYTE58= 000072
BYTE59= 000073
BYTE6 = 000006
BYTE60= 000074
BYTE61= 000075
BYTE62= 000076
BYTE63= 000077
BYTE64= 000100
BYTE65= 000101
BYTE66= 000102
BYTE67= 000103
BYTE68= 000104
BYTE69= 000105
BYTE7 = 000007
BYTE70= 000106
BYTE71= 000107
BYTE72= 000110
BYTE73= 000111
BYTE74= 000112
BYTE75= 000113
BYTE76= 000114
BYTE77= 000115
BYTE78= 000116
BYTE79= 000117
BYTE8 = 000010
BYTE80= 000120
BYTE81= 000121
BYTE82= 000122
BYTE83= 000123
BYTE84= 000124
BYTE85= 000125
BYTE86= 000126
BYTE87= 000127
BYTE88= 000130
BYTE89= 000131
BYTE9 = 000011
BYTE90= 000132
BYTE91= 000133
BYTE92= 000134
BYTE93= 000135
BYTE94= 000136
BYTE95= 000137
BYTE96= 000140
BYTE97= 000141
BYTE98= 000142
BYTE99= 000143
BYTVAL= 000144
CBKALL= 001000
CBKCLK= 000400
CNOBRE= 100000
COMXX = ****** GX
CONSOL= ****** GX
CPCCEN= 010000
CPREAD= 040000
CPWRTE= 020000
CSADRD= 000004
CSEQCI= 100000
CSOE  = 000040
CSWRTE= 000100
DBR.RD= 000001
DB$CPP= 001457
DB$SPT= 000026
DB$TPC= 000023
DISPGS= 100000
DMAAWR= 000005
DMARRD= 000003
DMARWR= 000004
ENBR  = 010000
ERR3  = ****** GX
ERR4  = ****** GX
ERR5  = ****** GX
FIND  = ****** GX
KILL  = ****** GX
LOC.EN= 000100
LOC.WA= 040000
LOC.WB= 100000
LOOP  = ****** GX
LOOPR = ****** GX
MAREN1= 000001
MAREN2= 004000
MARLOD= 010000
MAROUT= 000002
MAR.LO= 002000
MAR.OU= 000040
MBKALL= 001000
MBKCLK= 000400
MMADRD= 000100
MMLEFT= 000002
MMOE  = 000004
MMWRTE= 000010
MNOBRE= 100000
MREN1 = 000001
MREN2 = 020000
MSYN  = 000040
N     = 000144
ONCE  = ****** GX
PACK  = ****** GX
PLB   = 000010
PLC   = 000020
PLD   = 000030
PLRWR = 000200
PLR.EN= 000200
PRINT = ****** GX
QR$CR1= 176420
QR$CR2= 176422
QR$LBR= 176424
Q$ATTN= 000100
Q$BCL = 000001
Q$CCCP= 000040
Q$CHB = 000400
Q$CHRL= 000200
Q$CLR = 000040
Q$CNC = 030000
Q$CP  = 000060
Q$CPCC= 000010
Q$CP2 = 000260
Q$CSC = 010000
Q$CSEL= 000360
Q$CSET= 000002
Q$CSP = 020000
Q$DMA = 000001
Q$ENBK= 040000
Q$ENOP= 020000
Q$FAL = 004000
Q$FC  = 000045
Q$FO  = 000044
Q$FP  = 000046
Q$HBF = 000002
Q$ICP = 000006
Q$IHB = 000003
Q$IHRL= 000002
Q$IMRP= 000007
Q$LBD = 001000
Q$LBDP= 001001
Q$LBP = 000001
Q$LDCD= 000003
Q$LDMD= 000004
Q$LDPP= 002000
Q$LHP = 010000
Q$MNC = 140000
Q$MR  = 000052
Q$MRP = 000040
Q$MRP2= 000240
Q$MSC = 040000
Q$MSET= 000004
Q$MSP = 100000
Q$NCLK= 176000
Q$PP  = 000100
Q$PPSW= 000320
Q$PP2 = 000300
Q$QHLT= 000013
Q$QL  = 000043
Q$QLA = 000053
Q$QLB = 000054
Q$QLR = 000001
Q$QW  = 000042
Q$RDCD= 000005
Q$RDMD= 000006
Q$REBK= 001000
Q$RNC = 006000
Q$RSC = 004000
Q$RSET= 000010
Q$SM  = 100000
Q$SP  = 000120
Q$SP2 = 000340
REAA    000526RG    002
REBA    000542RG    002
RECA    000556RG    002
RECR    000702RG    002
REIA    000572RG    002
REIN    000516R     002
RELP    001042R     002
REMS    000716RG    002
REOA    000614RG    002
REPUT   001010R     002
REQR    000740RG    002
RESC    000660RG    002
RESW    000710RG    002
RETA    000636RG    002
RE1     000446RG    002
RE1LN = 000013 G
RE1TBL  000040RG    002
RE1X    001052R     002
RGO.EN= 000200
RGO.VA= 020000
RTNPT = ****** GX
SCAN  = ****** GX
SEQ.CI= 000010
STA     000224RG    002
STB     000234RG    002
STBT    000346RG    002
STC     000244RG    002
STCR    000416RG    002
STI     000276RG    002
STIN    000214R     002
STLP    000426R     002
STO     000322RG    002
STOR    000252R     002
STSC    000372RG    002
ST1     000114RG    002
ST1LN = 000010 G
ST1TBL  000000RG    002
ST1X    000436R     002
S$CLR = 000000
S$LO  = 000001
S$OB  = 000005
S$OR  = 000006
S$OX  = 000004
```

```
TREG-   M[illegible]0 M1110  27-MAR-80 15:39  PAGE 6-4
SYMBOL TA[illegible]
```

S$SR = 000007	T$IBAR = 000024	WORD15 = 000036	WORD44 = 000130	WORD73 = 000222
S$S1 = 000010	T$IBE = 020000	WORD16 = 000040	WORD45 = 000132	WORD74 = 000224
S$S2 = 000014	T$IBF = 040000	WORD17 = 000042	WORD46 = 000134	WORD75 = 000226
TD$CTR = 176370	T$ICD = 000040	WORD18 = 000044	WORD47 = 000136	WORD76 = 000230
TD$CTW = 176360	T$MODE = 004000	WORD19 = 000046	WORD48 = 000140	WORD77 = 000232
TD$INL = 004000	T$OB = 000036	WORD2 = 000004	WORD49 = 000142	WORD78 = 000234
TD$MEM = 000270	T$OBE = 004000	WORD20 = 000050	WORD5 = 000012	WORD79 = 000236
TD$OAR = 176344	T$OBF = 010000	WORD21 = 000052	WORD50 = 000144	WORD8 = 000020
TD$OTR = 176346	T$OBRA = 000034	WORD22 = 000054	WORD51 = 000146	WORD80 = 000240
TD$QRD = 000274	T$OBWA = 000032	WORD23 = 000056	WORD52 = 000150	WORD81 = 000242
TD$SW = 176376	T$OUTA = 100000	WORD24 = 000060	WORD53 = 000152	WORD82 = 000244
TD$TAR = 176372	T$RBDO = 000200	WORD25 = 000062	WORD54 = 000154	WORD83 = 000246
TD$TAW = 176362	T$RNB = 000040	WORD26 = 000064	WORD55 = 000156	WORD84 = 000250
TD$TDR = 176374	T$RSET = 040000	WORD27 = 000066	WORD56 = 000160	WORD85 = 000252
TD$TDW = 176364	T$SC = 000022	WORD28 = 000070	WORD57 = 000162	WORD86 = 000254
T$AD = 000020	T$SCLK = 020000	WORD29 = 000072	WORD58 = 000164	WORD87 = 000256
T$BA = 000002	T$SEG1 = 000000	WORD3 = 000006	WORD59 = 000166	WORD88 = 000260
T$BD = 000010	T$SEG2 = 000001	WORD30 = 000074	WORD6 = 000014	WORD89 = 000262
T$BSO = 100000	T$SEG3 = 000002	WORD31 = 000076	WORD60 = 000170	WORD9 = 000022
T$BT = 000020	T$SO = 000001	WORD32 = 000100	WORD61 = 000172	WORD90 = 000264
T$BTAR = 000030	T$UBUS = 100000	WORD33 = 000102	WORD62 = 000174	WORD91 = 000266
T$BTD = 002000	T$1CLK = 000400	WORD34 = 000104	WORD63 = 000176	WORD92 = 000270
T$CD = 000100	T$8BEN = 000020	WORD35 = 000106	WORD64 = 000200	WORD93 = 000272
T$CLK = 002000	UBD.IN = 000020	WORD36 = 000110	WORD65 = 000202	WORD94 = 000274
T$DISK = 000200	UNPK = ****** GX	WORD37 = 000112	WORD66 = 000204	WORD95 = 000276
T$DRD = 000004	WORD0 = 000000	WORD38 = 000114	WORD67 = 000206	WORD96 = 000300
T$EMEM = 010000	WORD1 = 000002	WORD39 = 000116	WORD68 = 000210	WORD97 = 000302
T$FSAA = 000000	WORD10 = 000024	WORD4 = 000010	WORD69 = 000212	WORD98 = 000304
T$FSAB = 000004	WORD11 = 000026	WORD40 = 000120	WORD7 = 000016	WORD99 = 000306
T$FSAC = 000014	WORD12 = 000030	WORD41 = 000122	WORD70 = 000214	WRDVAL = 000310
T$FSB2 = 000010	WORD13 = 000032	WORD42 = 000124	WORD71 = 000216	XTREAD = 001000
T$IB = 000026	WORD14 = 000034	WORD43 = 000126	WORD72 = 000220	XTWRTE = 000400

```
. ABS.  000000     000
        000000     001
TREG    001070     002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3266 WORDS  ( 13 PAGES)
DYNAMIC MEMORY:  3860 WORDS  ( 14 PAGES)
ELAPSED TIME:  00:00:45
TREG,TREG/-SP=[20,1]IM,[20,1]TREG
```

```
                                        .TITLE  TMEM
000000                                  .PSECT  TMEM
                                        .LIST   MEB
                                ;
                                ;
                                ;       TERM DETECTOR 'MANUAL' DEBUGGING AIDS
                                ;       MEMORY EXERCISES
                                ;
                                ;       COMMANDS:
                                ;       LD      LOAD MEMORIES
                                ;       PR      PRINT MEMORY CONTENTS
                                ;
                                ;       ONCE A COMMAND HAS BEEN EXECUTED (OR AN ERROR ENCOUNTERED)
                                ;       THIS MODULE RETURNS CONTROL TO THE MODULE TMAIN AT LOCATION
                                ;       'COMXX'.
                                ;
                                ;
                                ;       TABLE OF VALID MEMORY MNEMONICS AND THE ADDRESSES OF THEIR
                                ;       ASSOCIATED ROUTINES.
                                ;
000000                          LD1TBL::
000000  101     115                     .ASCII  /AM/            ;FSA A
000002  000332'                         .WORD   LDAM
000004  102     115                     .ASCII  /BM/            ;FSA B
000006  000366'                         .WORD   LDBM
000010  103     115                     .ASCII  /CM/            ;FSA C
000012  000422'                         .WORD   LDCM
000014  111     102                     .ASCII  /IB/            ;INPUT BUFFER
000016  000456'                         .WORD   LDIB
000020  117     102                     .ASCII  /OB/            ;OUTPUT BUFFER
000022  000654'                         .WORD   LDOB
000024  102     124                     .ASCII  /BT/            ;BYTE TRANSLATOR
000026  001112'                         .WORD   LDBT
        000006                  LD1LN   ==      <.-LD1TBL>/4
                                ;
                                ;       PRINT
                                ;
000030                          PR1TBL::
000030  101     115                     .ASCII  /AM/            ;FSA A
000032  001650'                         .WORD   PRAM
000034  102     115                     .ASCII  /BM/            ;FSA B
000036  001704'                         .WORD   PRBM
000040  103     115                     .ASCII  /CM/            ;FSA C
000042  001740'                         .WORD   PRCM
000044  111     102                     .ASCII  /IB/            ;INPUT BUFFER
000046  001774'                         .WORD   PRIB
000050  117     102                     .ASCII  /OB/            ;OUTPUT BUFFER
000052  002072'                         .WORD   PROB
000054  102     124                     .ASCII  /BT/            ;BYTE TRANSLATOR
000056  002210'                         .WORD   PRBT
        000006                  PR1LN   ==      <.-PR1TBL>/4
                                ;
                                ;
                                ;       LOAD
                                ;
                                ;
000060                          LD1::
```

```
 58 000060  004767  000000G·····              JSR·    PC,FIND·        ;LOCATE·MEMORY·MNEMONIC·IN·COMMAND·LINE·
 59 000064  103004                            BCC·    1$              ;OK, CONTINUE·
 60 000066  004767  000000G·                  JSR·    PC,ERR3         ;'MISSING·OPERAND'
 61 000072·  000167  001246                   JMP·    LDiX            ;EXIT·
 62·                                  ;
 63 000076  012700  000006           1$:      MOV·    #LDiLN,R0       ;NUMBER·OF·TABLE·ENTRIES·
 64 000102·  012702·  000000'                 MOV·    #LDiTBL,R2·     ;POINT·TO·TABLE·
 65 000106  004767  000000G·                  JSR·    PC,SCAN·        ;MATCH·AGAINST·COMMAND·LINE·
 66 000112·  103004                           BCC·    2$              ;MATCH·WAS·MADE·
 67 000114  004767  000000G·                  JSR     PC,ERR6         ;'INVALID·MEMORY·MNEMONIC'
 68 000120  000167  001220                   JMP·    LDiX
 69                                   ;
 70                                   ;       SAVE·POINTER TO·ROUTINE·ASSOCIATED·WITH·THE·MEMORY·MNEMONIC·
 71                                   ;       SCAN·THE·REMAINDER·OF·THE·COMMAND·LINE·FOR CONTROL·INFORMATION:
 72·                                  ;
 73                                   ;       1. START·ADDRESS·ONLY, REPEAT·PROMPT·FOR·EACH·MEMORY LOCATION·
 74                                   ;          FROM·THE·START·ADDRESS·FORWARD. NB: END OF·MEMORY (AS·DETECTED·
 75                                   ;          BY·THE·QMAIN·SUBROUTINE·PDATA) OR·A·<CR> RESPONSE TO·THE·PROMPT·
 76                                   ;          WILL·TERMINATE·THE·LOAD.
 77                                   ;          >LD·OB·0
 78                                   ;       2. START·ADDRESS, LOOP INDICATOR. LOOP·ON·THE·LOADING·OF·
 79                                   ;          THIS·ONE·MEMORY·LOCATION·ONLY.
 80                                   ;          >LD·OB·0 L·
 81                                   ;       3. START·ADDRESS, END·ADDRESS, NO·LOOP, ISSUE·ONLY·ONE·
 82·                                  ;          PROMPT·AND·FILL·MEMORY·(BETWEEN·START·AND·END·ADDRESSES)
 83                                   ;          WITH·THIS VALUE.
 84                                   ;          >LD·OB·0 6
 85                                   ;       4. START·ADDRESS, END·ADDRESS, LOOP·INDICATOR, ISSUE ONLY·ONE·
 86                                   ;          PROMPT·AND·LOOP·ON·THE·LOADING·OF·MEMORY·(BETWEEN START·
 87                                   ;          AND·END·ADDRESSES) WITH·THIS·VALUE.
 88                                   ;          >LD·OB·0 6 L·
 89                                   ;
 90                                   ;
 91 000124  010167  000000G·          2$:      MOV·    R1,RTNPT·       ;SAVE·POINTER·
 92·000130  004767  000000G·                   JSR·    PC,FIND·        ;LOCATE·START·ADDR IN·COMMAND·LINE·
 93 000134  103004                            BCC·    3$              ;OK, CONTINUE·
 94 000136  004767  000000G·                   JSR     PC,ERR3         ;'MISSING·OPERAND'
 95 000142·  000167  001176                    JMP     LDiX            ;EXIT·
 96 000146  004767  000000G·          3$:      JSR·    PC,PACK·        ;CONVERT COMMAND·LINE·VALUE·TO·BINARY·
 97 000152·  103004                            BCC·    4$              ;CONVERSION·SUCCESSFUL·
 98 000154  004767  000000G·                   JSR·    PC,ERR4         ;'INVALID·NUMERIC·VALUE'
 99 000160  000167  001160                    JMP·    LDiX
100                                   ;
101 000164  016767  000000G·000000G· 4$:      MOV·    BINWD,MSTRT·    ;SAVE·LOADING·START·ADDRESS·
102·000172·  016767  000000G·000000G·         MOV·    BINWD,MSTR2·    ;SAVE·IT TWICE·(FOR·REFRESH·ON·LOOP)
103 000200  012767  177777  000000G·         MOV·    #-1,MEND·       ;INIT·END·ADDRESS·
104                                  ;
105·                                 ;       START·ADDRESS·HAS·BEEN FOUND, SCAN·FOR·END ADDRESS·
106                                  ;       OR LOOP·INDICATOR·(CONDITIONS·2, 3 ABOVE), IF·THERE·
107                                  ;       IS NOTHING·FURTHER·IN·THE·COMMAND·LINE, CONDITION·
108                                  ;       1 IS·IN·EFFECT.
109                                  ;
110 000206  004767  000000G·                  JSR·    PC,FIND·        ;SCAN·COMMAND·LINE
111 000212  103004                            BCC·    5$              ;SOMETHING·THERE·
112·000214  052767  000000G·000000G·         BIS·    #RP,BASE·       ;SIGNAL·TO·REPEAT·PROMPT·
113 000222·  000437                           BR      9$              ;JUMP·TO RTN·
114                                  ;
```

```
000224  122711  000114           5$:     CMPB    #'L,(R1)        ;LOOP INDICATOR
000230  001006                           BNE     6$              ;NO, MUST BE UPPER ADDRESS
000232  016767  000000G 000000G          MOV     MSTRT,MEND      ;LOWER ADDRESS BECOMES UPPER ADDRESS
000240  004767  000000G                  JSR     PC,HANG         ;HOW TO STOP LOOP
000244  000426                           BR      9$              ;JUMP TO RTN
                                 ;
000246  004767  000000G          6$:     JSR     PC,PACK         ;CONVERT UPPER ADDRESS
000252  103004                           BCC     7$              ;OK, CONTINUE
000254  004767  000000G                  JSR     PC,ERR4         ;'INVALID NUMERIC VALUE'
000260  000167  001060                   JMP     LD1X            ;AND EXIT
                                 ;
                                 ;       SAVE END ADDRESS (BINARY)
                                 ;       CHECK FOR LOOP INDICAOR AFTER END ADDR (CONDITION 4).
                                 ;
000264  016767  000000G 000000G  7$:     MOV     BINWD,MEND      ;SET UP ENDING ADDRESS
000272  004767  000000G                  JSR     PC,FIND         ;CHECK FOR LOOP INDICATOR
000276  103411                           BCS     9$              ;NO LOOP
000300  122711  000114                   CMPB    #'L,(R1)        ;CORRECT INDICATOR
000304  001404                           BEQ     8$              ;YES, CONTINUE
000306  004767  000000G                  JSR     PC,ERR1         ;NO, PUT OUT MESSAGE
000312  000167  001026                   JMP     LD1X            ;AND EXIT
000316  004767  000000G          8$:     JSR     PC,HANG         ;HOW TO STOP LOOP
                                 ;
000322  016701  000000G          9$:     MOV     RTNPT,R1        ;POINT TO ROUTINE
000326  000171  000000                   JMP     @(R1)           ;JUMP TO ROUTINE
                                 ;
                                 ;
                                 ;       LOAD FSA A
                                 ;
000332                           LDAM::
000332  012767  000000  000000G          MOV     #T$FSAA,APLACE  ;SELECT FSA A FOR SUBRTN
000340  016746  000000G                  MOV     AHIGH,-(SP)     ;UPPER MEMORY LIMIT
000344  016746  000000G                  MOV     ALOW,-(SP)      ;LOWER LIMIT
000350  004767  000000G                  JSR     PC,BUFFSA       ;CHECK ADDRESS LIMITS, ETC
000354  103402                           BCS     1$              ;FAILED CHECK
000356  004767  000000G                  JSR     PC,LDCNTL       ;LOAD FSA
000362  000167  000756           1$:     JMP     LD1X            ;AND EXIT
                                 ;
                                 ;       LOAD FSA B
                                 ;
000366                           LDBM::
000366  012767  000004  000000G          MOV     #T$FSAB,APLACE  ;SELECT FSA B FOR SUBRTN
000374  016746  000000G                  MOV     BHIGH,-(SP)     ;UPPER MEMORY LIMIT
000400  016746  000000G                  MOV     BLOW,-(SP)      ;LOWER LIMIT
000404  004767  000000G                  JSR     PC,BUFFSA       ;CHECK ADDRESS LIMITS, ETC
000410  103402                           BCS     1$              ;FAILED CHECK
000412  004767  000000G                  JSR     PC,LDCNTL       ;LOAD FSA
000416  000167  000722           1$:     JMP     LD1X            ;AND EXIT
                                 ;
                                 ;       LOAD FSA C
                                 ;
000422                           LDCM::
000422  012767  000014  000000G          MOV     #T$FSAC,APLACE  ;SELECT FSA C FOR SUBRTN
000430  016746  000000G                  MOV     CHIGH,-(SP)     ;MOVE UPPER MEMORY LIMIT
000434  016746  000000G                  MOV     CLOW,-(SP)      ;LOWER LIMIT
000440  004767  000000G                  JSR     PC,BUFFSA       ;CHECK ADDRESS LIMITS, ETC
000444  103402                           BCS     1$              ;FAILED CHECK
```

```
172·000446  004767  000000G·                 JSR·    PC,LDCNTL·      ;LOAD·FSA·
173 000452· 000167  000666                  1$:     JMP     LD1X            ;AND·EXIT·
174                                         ;
175                                         ;
176                                         ;
177                                         ;       LOAD·INPUT·BUFFER·
178                                         ;
179 000456                                  LDIB::
180 000456  012767  000200·  000000G·               MOV·    #T$DISK,APLACE·  ;SIMULATE·DISK·
181 000464  016746  000000G·                        MOV·    IHIGH,-(SP)      ;SUPPLY·UPPER·MEMORY·LIMIT·
182·000470  016746  000000G·                        MOV·    ILOW,-(SP)       ;LOWER·LIMIT·
183 000474  004767  000000G·                        JSR·    PC,BUFS2·        ;PREPARE·TO·LOAD·
184 000500  103002·                                 BCC·    100$             ;NO·ERRORS,·CONTINUE·
185 000502· 000167  000636                          JMP·    LD1X
186                                         ;
187 000506  032767  000000G·000000G·       100$:   BIT·    #RP,BASE·        ;REPEAT·PROMPT·?·
188 000514  001420                                  BEQ·    2$               ;NO,·ONCE·ONLY·
189 000516  004767  000000G·                 1$:     JSR·    PC,PDATA·        ;READ·DATA·FROM·COMMAND·LINE
190 000522· 103002·                                 BCC·    10$              ;NOT·END·OF·MEMORY
191 000524  000167  000614                          JMP·    LD1X             ;IF·END·OR·ERROR,·EXIT·
192·000530  102002·                         10$:    BVC·    20$              ;<CR>·RESPONSE·TO·PROMPT·
193 000532  000167  000606                          JMP·    LD1X             ;YES,·EXIT·
194 000536  016702· 000000G·                20$:    MOV·    MSTR2,R2·        ;LOAD·IB·ADDRESS
195 000542· 016712· 000000G·                        MOV·    DATA1,(R2)       ;LOAD·
196 000546  066767  000000G·000000G·               ADD·    INCVAL,MSTR2·    ;ADVANCE·ADDRESS·
197 000554  000760                                  BR      1$               ;REPEAT·
198                                         ;
199                                         ;       PROMPT·ONCE·THEN·FILL·MEMORY·
200                                         ;
201 000556  004767  000000G·                 2$:     JSR·    PC,PDATA·        ;PROMPT·ONCE·
202·000562· 103002·                                 BCC·    30$              ;NOT·END·OF·MEMORY·OR·ERROR·
203 000564  000167  000554                          JMP·    LD1X             ;IF·YES,·EXIT·
204 000570  102004                          30$:    BVC·    40$              ;<CR>·RESPONSE·TO·PROMPT·
205 000572· 004767  000000G·                        JSR·    PC,ERR3          ;YES,·MISSING·OPERAND·
206 000576  000167  000542                          JMP·    LD1X
207 000602· 016702· 000000G·                40$:    MOV·    MSTR2,R2·        ;LOAD·IB·ADDRESS·
208 000606  016712· 000000G·                        MOV·    DATA1,(R2)
209 000612· 066767  000000G·000000G·               ADD·    INCVAL,MSTR2·    ;ADVANCE·ADDRESS·
210 000620  026767  000000G·000000G·               CMP·    MSTR2,MEND·      ;HAS·UPPER·MEMORY·LIMIT·BEEN·REACHED·
211 000626  101765                                  BLOS·   40$              ;NO,·CONTINUE·
212·000630  032767  000000G·000000G·               BIT·    #LOOP,BASE·      ;REPEAT·COMMAND·
213 00063E  001002·                                 BNE·    50$              ;YES,·CONTINUE·
214 000640  000167  000500                          JMP·    LD1X             ;ELSE·EXIT·
215 000644  016767  000000G·000000G·50$:    MOV·    MSTRT,MSTR2·     ;REINITIALIZE·ADDRESS·
216 000652· 000753                                  BR      40$
217                                         ;
218                                         ;
219                                         ;       LOAD·OUTPUT·BUFFER·
220                                         ;
221 000654                                  LDOB::
222·000654  012767  000040  000000G·               MOV·    #T$ICD,APLACE·   ;SET·INTERFACE·AND·CONTROL·FOR·SUBRTN·
223 000662· 016746  000000G·                        MOV·    OHIGH,-(SP)      ;SUPPLY·UPPER·MEMORY·LIMIT·
224 000666  016746  000000G·                        MOV·    OLOW,-(SP)       ;LOWER·LIMIT·
225 000672· 004767  000000G·                        JSR·    PC,BUFS2·        ;PREPARE·FOR·LOAD·
226 000676  103002·                                 BCC·    100$             ;OK,·CONTINUE·
227 000700  000167  000440                          JMP·    LD1X             ;ERROR,·EXIT·
228                                         ;
```

```
000704  032767  000000G 000000G 100$:   BIT     #RP,BASE        ;REPEAT PROMPT ?
000712  001430                          BEQ     2$              ;NO, ONCE ONLY
000714  004767  000000G         1$:     JSR     PC,PDATA        ;READ DATA FROM COMMAND LINE
000720  103002                          BCC     10$             ;NOT END OF MEMORY
000722  000167  000416                  JMP     LDIX            ;IF END OR ERROR, EXIT
000726  102002                  10$:    BVC     20$             ;<CR> RESPONSE TO PROMPT
000730  000167  000410                  JMP     LDIX            ;YES, EXIT
000734  012767  000032  176362  20$:    MOV     #T$OBWA,TD$TAW
000742  016767  000000G 176364          MOV     MSTR2,TD$TDW
000750  012767  000036  176362          MOV     #T$OB,TD$TAW
000756  016767  000000G 176364          MOV     DATA1,TD$TDW
000764  066767  000000G 000000G         ADD     INCVAL,MSTR2    ;ADVANCE ADDRESS
000772  000750                          BR      1$              ;REPEAT
                                ;
                                ;       PROMPT ONCE THEN FILL MEMORY
                                ;
000774  004767  000000G         2$:     JSR     PC,PDATA        ;PROMPT ONCE
001000  103002                          BCC     30$             ;NOT END OF MEMORY OR ERROR
001002  000167  000336                  JMP     LDIX            ;IF YES, EXIT
001006  102004                  30$:    BVC     40$             ;<CR> RESPONSE TO PROMPT ?
001010  004767  000000G                 JSR     PC,ERR3         ;YES, MISSING OPERAND
001014  000167  000324                  JMP     LDIX
001020  012767  000032  176362  40$:    MOV     #T$OBWA,TD$TAW
001026  016767  000000G 176364          MOV     MSTR2,TD$TDW
001034  012767  000036  176362          MOV     #T$OB,TD$TAW
001042  016767  000000G 176364          MOV     DATA1,TD$TDW
001050  066767  000000G 000000G         ADD     INCVAL,MSTR2    ;ADVANCE ADDRESS
001056  026767  000000G 000000G         CMP     MSTR2,MEND      ;HAS UPPER MEMORY LIMIT BEEN REACHED
001064  101755                          BLOS    40$             ;NO, CONTINUE
001066  032767  000000G 000000G         BIT     #LOOP,BASE      ;REPEAT COMMAND
001074  001002                          BNE     50$             ;YES, CONTINUE
001076  000167  000242                  JMP     LDIX            ;ELSE EXIT
001102  016767  000000G 000000G 50$:    MOV     MSTRT,MSTR2     ;REINITIALIZE ADDRESS
001110  000743                          BR      40$
                                ;
                                ;
                                ;       LOAD BYTE TRANSLATOR
                                ;
001112                          LDBT::
001112  012767  000040  000000G         MOV     #T$ICD,APLACE   ;SET INTERFACE AND CONTROL FOR SUBRTN
001120  016746  000000G                 MOV     THIGH,-(SP)     ;SUPPLY MEMORY UPPER LIMIT
001124  016746  000000G                 MOV     TLOW,-(SP)      ;LOWER LIMIT
001130  004767  000000G                 JSR     PC,BUFSET       ;PREPARE FOR LOAD
001134  103503                          BCS     LDIX            ;ERROR
                                ;
001136  032767  000000G 000000G         BIT     #RP,BASE        ;REPEAT PROMPT ?
001144  001430                          BEQ     2$              ;NO, PROMPT ONLY ONCE
001146  004767  000000G         1$:     JSR     PC,PDATA        ;READ DATA FROM COMMAND LINE
001152  103002                          BCC     10$             ;VALID DATA PRESENT
001154  000167  000164                  JMP     LDIX            ;IF END OF MEMORY OR ERROR, EXIT
001160  102002                  10$:    BVC     20$             ;<CR> RESPONSE TO PROMPT
001162  000167  000156                  JMP     LDIX            ;YES, EXIT
001166  012767  000030  176362  20$:    MOV     #T$BTAR,TD$TAW  ;SELECT BYTE TRANS REG
001174  016767  000000G 176364          MOV     MSTR2,TD$TDW    ;MOVE ADDR TO TRANSFER REG
001202  012767  000020  176362          MOV     #T$BT,TD$TAW    ;SELECT BYTE TRANS MEMORY
001210  016767  000000G 176364          MOV     DATA1,TD$TDW    ;LOAD BYTE TRANS
001216  066767  000000G 000000G         ADD     INCVAL,MSTR2    ;ADVANCE ADDRESS
```

```
286 001224  000750                          BR      1$              ;REPEAT·
287                                 ;
288                                 ;       PROMPT·ONCE·THEN·FILL·MEMORY·
289                                 ;
290 001226  004767  000000G·        2$:     JSR·    PC,PDATA·       ;PROMPT·ONCE·
291 001232· 103002·                         BCC·    30$             ;NOT·END·OF·MEMORY·OR·ERROR·
292·001234  000167  000104                  JMP·    LD1X            ;IF·YES, EXIT·
293 001240  102004                  30$:    BVC·    40$             ;<CR> RESPONSE·TO·PROMPT·?·
294 001242· 004767  000000G·                JSR·    PC,ERR3         ;YES, MISSING·OPERAND·
295 001246  000167  000072                  JMP·    LD1X
296 001252· 012767  000030  176372· 40$:    MOV·    #T$BTAR,TD$TAR  ;SELECT·BYTE·TRANS REG·
297 001260  016767  000000G·176364         MOV·    MSTR2,TD$TDW    ;MOVE·ADDR·TO·TRANSFER·REG·
298 001266  012767  000020  176362·        MOV·    #T$BT,TD$TAW·   ;SELECT·BYTE·TRANS MEMORY·
299 001274  016767  000000G·176364         MOV·    DATA1,TD$TDW·   ;LOAD·BYTE·TRANS·
300 001302· 066767  000000G·000000G·        ADD·    INCVAL,MSTR2·   ;ADVANCE ADDRESS·
301 001310  026767  000000G·000000G·        CMP·    MSTR2,MEND·     ;HAS·UPPER·MEMORY·LIMIT·BEEN REACHED·
302·001316  101755                          BLOS·   40$             ;NO, CONTINUE·
303 001320  032767  000000G·000000G·        BIT·    #LOOP,BASE·     ;REPEAT·COMMAND·
304 001326  001002·                         BNE·    50$             ;YES, CONTINUE·
305 001330  000167  000010··                JMP·    LD1X            ;ELSE·EXIT·
306 001334  016767  000000G·000000G·50$:    MOV·    MSTRT,MSTR2·    ;REINITIALIZE·ADDRESS·
307 001342· 000743                          BR      40$
308                                 ;
309                                 ;
310 001344                          LD1X:
311 001344  042767  000000G·000000G·        BIC·    #RP,BASE·       ;CLEAR·PROMPT·FLAG
312·001352· 004767  000000G·                JSR·    PC,KILL·        ;KILL·AST·(IF·THERE·WAS·ONE)
313 001356  000167  000000G··               JMP·    COMXX·
```

```
                                        ;
                                        ;
                                        ;
                                        ;       PRINT.
                                        ;       PERFROM SECOND LEVEL PARSING.
                                        ;       EG. IN THE COMMAND:
                                        ;       >PR OB 0
                                        ;       PARSE THE 'OB'
                                        ;
                                        ;
001362                                  PR1::
001362  004767  000000G                         JSR     PC,FIND         ;LOCATE MEMORY MNEMONIC IN COMMAND LINE.
001366  103004                                  BCC     1$              ;OK, CONTINUE.
001370  004767  000000G                         JSR     PC,ERR3         ;'MISSING OPERAND'
001374  000167  000726                         JMP     PR1X            ;EXIT.
                                        ;
                                        ;       SCAN THE TABLE OF VALID MEMORY MNEMONICS AND ASSOCIATED
                                        ;       ROUTINE ADDRESSES.
                                        ;
001400  012700  000006                  1$:     MOV     #PR1LN,R0       ;NUMBER OF TABLE ENTRIES.
001404  012702  000030'                         MOV     #PR1TBL,R2      ;POINT TO TABLE.
001410  004767  000000G                         JSR     PC,SCAN         ;MATCH AGAINST COMMAND LINE.
001414  103004                                  BCC     2$              ;MATCH WAS MADE.
001416  004767  000000G                         JSR     PC,ERR6         ;'INVALID MEMORY MNEMONIC'
001422  000167  000700                         JMP     PR1X
                                        ;
                                        ;       SAVE POINTER TO ROUTINE ASSOCIATED WITH THE MEMORY MNEMONIC,
                                        ;       SCAN THE REMAINDER OF THE COMMAND LINE FOR CONTROL INFORMATION:
                                        ;
                                        ;       1. START ADDRESS ONLY, PRINT ONLY ONE MEMORY LOCATION.
                                        ;          >PR OB 0
                                        ;       2. START ADDRESS, LOOP INDICATOR, LOOP ON THE READING OF
                                        ;          THIS ONE MEMORY LOCATION ONLY, PRINT THE CONTENTS ONLY
                                        ;          ONCE.
                                        ;          >PR OB 0 L.
                                        ;       3. START ADDRESS, END ADDRESS, NO LOOP, PRINT THE CONTENTS
                                        ;          MEMORY BETWEEN THE START ADDRESS AND THE END ADDRESS:
                                        ;          >PR OB 0 6
                                        ;       4. START ADDRESS, END ADDRESS, LOOP INDICATOR, READ THE CONTENTS
                                        ;          OF MEMORY BETWEEN THE START AND END ADDRESSES, HOWEVER,
                                        ;          PRINT ONLY THE CONTENTS OF MEMORY AT THE START ADDRESS.
                                        ;          >PR OB 0 6 L.
                                        ;
                                        ;
001426  010167  000000G                 2$:     MOV     R1,RTNPT        ;SAVE POINTER.
001432  004767  000000G                         JSR     PC,FIND         ;LOCATE START ADDR IN COMMAND LINE.
001436  103004                                  BCC     3$              ;OK, CONTINUE.
001440  004767  000000G                         JSR     PC,ERR3         ;'MISSING OPERAND'
001444  000167  000656                         JMP     PR1X            ;EXIT.
001450  004767  000000G                 3$:     JSR     PC,PACK         ;CONVERT COMMAND LINE VALUE TO BINARY.
001454  103004                                  BCC     4$              ;CONVERSION SUCCESSFUL.
001456  004767  000000G                         JSR     PC,ERR4         ;'INVALID NUMERIC VALUE'
001462  000167  000640                         JMP     PR1X
                                        ;
001466  016767  000000G 000000G         4$:     MOV     BINWD,MSTRT     ;SAVE LOADING START ADDRESS.
001474  016767  000000G 000000G                 MOV     BINWD,MSTR2     ;SAVE IT TWICE.
001502  012767  177777  000000G                 MOV     #-1,MEND        ;INIT END ADDRESS.
```

```
                                        ;
                                        ;       START·ADDRESS·HAS·BEEN FOUND. SCAN·FOR·END ADDRESS·OR·
                                        ;       LOOP·INDICATOR·(CONDITIONS·2, 3 ABOVE). IF THERE·IS·
                                        ;       NOTHING·FURTHER·IN·THE COMMAND·LINE, CONDITION·1 IS·
                                        ;       IN EFFECT.
                                        ;
001510  004767  000000G·                        JSR·    PC,FIND·        ;SCAN·COMMAND·LINE
001514  103004                                  BCC·    5$              ;SOMETHING·THERE·
001516  016767  000000G·000000G·                MOV·    MSTRT,MEND·     ;SET·END ADDR·=·START·ADDR·
001524  000445                                  BR      9$              ;JUMP·TO RTN·
                                        ;
001526  122711  000114                5$:     CMPB·   #'L,(R1)        ;LOOP·INDICATOR·
001532  001011                                BNE·    6$              ;NO, MUST·BE·UPPER ADDRESS·
001534  052767  000000G·000000G·              BIS·    #OUT,BASE·      ;SET·FLAG·FOR·OUTPUT·CONTROL
001542  016767  000000G·000000G·              MOV·    MSTRT,MEND·     ;LOWER·ADDRESS·BECOMES·UPPER ADDRESS·
001550  004767  000000G·                      JSR·    PC,HANG·        ;HOW·TO·STOP·LOOP·
001554  000431                                BR      9$              ;JUMP·TO RTN·
                                        ;
001556  004767  000000G·                6$:     JSR·    PC,PACK·        ;CONVERT UPPER·ADDRESS
001562  103004                                  BCC·    7$              ;OK, CONTINUE·
001564  004767  000000G·                        JSR·    PC,ERR4         ;'INVALID·NUMERIC·VALUE'
001570  000167  177550                        JMP·    LD1X            ;AND·EXIT·
                                        ;
                                        ;       SAVE·END·ADDRESS·(BINARY)
                                        ;       CHECK·FOR·LOOP·INDICATOR·AFTER·END·ADDRESS (CONDITION·4).
                                        ;
001574  016767  000000G·000000G·7$:     MOV·    BINWD,MEND·     ;SET·UP·ENDING·ADDRESS·
001602  004767  000000G·                      JSR·    PC,FIND·        ;CHECK·FOR·LOOP·INDICATOR·
001606  103414                                BCS·    9$              ;NO·LOOP
001610  122711  000114                        CMPB·   #'L,(R1)        ;CORRECT LOOP·INDICATOR·
001614  001404                                BEQ·    8$              ;YES, CONTINUE·
001616  004767  000000G·                      JSR·    PC,ERR1
001622  000167  000500                        JMP·    PR1X            ;EXIT·
001626  052767  000000G·000000G·8$:     BIS·    #OUT,BASE·      ;SET·OUTPUT·CONTROL·
001634  004767  000000G·                      JSR·    PC,HANG·        ;HOW·TO·STOP·LOOP·
                                        ;
001640  016701  000000G·                9$:     MOV·    RTNPT,R1        ;POINT·TO·ROUTINE·
001644  000171  000000                        JMP·    @(R1)           ;JUMP·TO ROUTINE·
                                        ;
                                        ;
                                        ;       PRINT·FROM·FSA·A·
                                        ;
001650                                  PRAM::
001650  012767  000000· 000000G·               MOV·    #T$FSAA,APLACE  ;SELECT·FSA·A·FOR·SUBRTN·
001656  016746  000000G·                      MOV·    AHIGH,-(SP)     ;UPPER·MEMORY·LIMIT·
001662  016746  000000G·                      MOV·    ALOW,-(SP)      ;LOWER·LIMIT·
001666  004767  000000G·                      JSR·    PC,BUFFSA·      ;CHECK·ADDRESS·AGAINST·MEMORY·LIMITS·
001672  103402·                              BCS·    1$              ;CHECK·FAILED·
001674  004767  000000G·                      JSR·    PC,PRCNTL·      ;PRINT·MEMORY·
001700  000167  000422                1$:     JMP·    PR1X            ;AND·EXIT·
                                        ;
                                        ;       PRINT·FROM·FSA·B·
                                        ;
001704                                  PRBM::
001704  012767  000004  000000G·               MOV·    #T$FSAB,APLACE  ;SELECT·FSA·B·FOR·SUBRTN·
001712  016746  000000G·                      MOV·    BHIGH,-(SP)     ;UPPER·MEMORY·LIMIT·
001716  016746  000000G·                      MOV·    BLOW,-(SP)      ;LOWER·LIMIT·
```

```
001722  004767  000000G               JSR   PC,BUFFSA        ;CHECK ADDRESS AGAINST MEMORY LIMITS
001726  103402                        BCS   1$               ;CHECK FAILED
001730  004767  000000G               JSR   PC,PRCNTL        ;PRINT MEMORY
001734  000167  000366        1$:     JMP   PR1X             ;AND EXIT
                              ;
                              ;       PRINT FROM FSA C
                              ;
001740                        PRCM::
001740  012767  000014  000000G       MOV   #T$FSAC,APLACE   ;SELECT FSA C FOR SUBRTN
001746  016746  000000G               MOV   CHIGH,-(SP)      ;MOVE UPPER MEMORY LIMIT
001752  016746  000000G               MOV   CLOW,-(SP)       ;LOWER LIMIT
001756  004767  000000G               JSR   PC,BUFFSA        ;CHECK ADDRESS AGAINST MEMORY LIMITS
001762  103402                        BCS   1$               ;CHECK FAILED
001764  004767  000000G               JSR   PC,PRCNTL        ;PRINT MEMORY
001770  000167  000332        1$:     JMP   PR1X             ;AND EXIT
                              ;
                              ;
                              ;       PRINT FROM INPUT BUFFER
                              ;
001774                        PRIB::
001774  012767  000200  000000G       MOV   #T$DISK,APLACE   ;SELECT DISK SIMULATION
002002  016746  000000G               MOV   IHIGH,-(SP)      ;SUPPLY UPPER MEMORY LIMIT
002006  016746  000000G               MOV   ILOW,-(SP)       ;LOWER LIMIT
002012  004767  000000G               JSR   PC,BUFS2         ;PREPARE TO LOAD
002016  103543                        BCS   PR1X             ;ERROR ENCOUNTERED
                              ;
002020  016702  000000G       1$:     MOV   MSTR2,R2         ;LOAD IB ADDRESS
002024  011267  000000G               MOV   (R2),DATA1       ;READ IB
002030  004767  000000G               JSR   PC,PRDATA        ;CALL PRINT SUBROUTINE
                              ;
002034  066767  000000G 000000G       ADD   INCVAL,MSTR2     ;ADVANCE ADDRESS
002042  026767  000000G 000000G       CMP   MSTR2,MEND       ;HAS UPPER MEMORY LIMIT BEEN REACHED
002050  101763                        BLOS  1$               ;NO, CONTINUE
002052  032767  000000G 000000G       BIT   #LOOP,BASE       ;LOOP ON ?
002060  001522                        BEQ   PR1X             ;NO, EXIT
002062  016767  000000G 000000G       MOV   MSTRT,MSTR2      ;REINIT ADDRESS
002070  000753                        BR    1$               ;START OVER
                              ;
                              ;
                              ;       PRINT FROM OUTPUT BUFFER
                              ;
002072                        PROB::
002072  012767  000040  000000G       MOV   #T$ICD,APLACE    ;SELECT INTERFACE AND CONTROL FOR SUBRTN
002100  016746  000000G               MOV   OHIGH,-(SP)      ;SUPPLY UPPER MEMORY LIMIT
002104  016746  000000G               MOV   OLOW,-(SP)       ;LOWER LIMIT
002110  004767  000000G               JSR   PC,BUFS2         ;PREPARE FOR LOAD
002114  103504                        BCS   PR1X             ;ERROR
                              ;
002116  012767  000034  176362 1$:    MOV   #T$OBRA,TD$TAW   ;SELECT OUTPUT BUFFER WRITE REG
002124  016767  000000G 176364        MOV   MSTR2,TD$TDW     ;MOVE ADDR TO TRANSFER REG
002132  012767  000036  176372        MOV   #T$OB,TD$TAR     ;PREPARE FOR LOAD
002140  016767  176374  000000G       MOV   TD$TDR,DATA1     ;LOAD
002146  004767  000000G               JSR   PC,PRDATA        ;CALL PRINT SUBROUTINE
                              ;
002152  066767  000000G 000000G       ADD   INCVAL,MSTR2     ;ADVANCE ADDRESS
002160  026767  000000G 000000G       CMP   MSTR2,MEND       ;HAS UPPER MEMORY LIMIT BEEN REACHED
002166  101753                        BLOS  1$               ;NO, CONTINUE
```

```
 486 002170  032767  000000G 000000G          BIT     #LOOP,BASE      ;LOOP ON ?
 487 002176  001453                          BEQ     PR1X            ;NO, EXIT
 488 002200  016767  000000G 000000G          MOV     MSTRT,MSTR2     ;REINIT ADDRESS
 489 002206  000743                          BR      1$              ;START OVER
 490                                 ;
 491                                 ;
 492                                 ;       PRINT FROM BYTE TRANSLATOR
 493                                 ;
 494 002210                          PRBT::
 495 002210  012767  000040  000000G          MOV     #T$ICD,APLACE   ;SELECT INTERFACE AND CNTL FOR SUBRTN
 496 002216  016746  000000G                  MOV     THIGH,-(SP)     ;SUPPLY MEMORY UPPER LIMIT
 497 002222  016746  000000G                  MOV     TLOW,-(SP)      ;LOWER LIMIT
 498 002226  004767  000000G                  JSR     PC,BUFSET       ;PREPARE FOR LOAD
 499 002232  103435                          BCS     PR1X            ;ERROR
 500                                 ;
 501 002234  012767  000030  176362  1$:     MOV     #T$BTAR,TD$TAW  ;SELECT BYTE TRANS REG
 502 002242  016767  000000G 176364          MOV     MSTR2,TD$TDW    ;MOVE ADDR TO TRANSFER REG
 503 002250  012767  000020  176372          MOV     #T$BT,TD$TAR    ;SELECT BYTE TRANS MEMORY
 504 002256  016767  176374  000000G          MOV     TD$TDR,DATA1    ;LOAD BYTE TRANS
 505 002264  004767  000000G                  JSR     PC,PRDATA       ;CALL PRINT SUBROUTINE
 506                                 ;
 507 002270  066767  000000G 000000G          ADD     INCVAL,MSTR2    ;ADVANCE ADDRESS
 508 002276  026767  000000G 000000G          CMP     MSTR2,MEND      ;HAS UPPER MEMORY LIMIT BEEN REACHED
 509 002304  101753                          BLOS    1$              ;NO, CONTINUE
 510 002306  032767  000000G 000000G          BIT     #LOOP,BASE      ;LOOP ON ?
 511 002314  001404                          BEQ     PR1X            ;NO, EXIT
 512 002316  016767  000000G 000000G          MOV     MSTRT,MSTR2     ;REINIT ADDRESS
 513 002324  000743                          BR      1$              ;START OVER
 514                                 ;
 515                                 ;
 516 002326                          PR1X:
 517 002326  042767  000000C 000000G          BIC     #<OUT+ONCE>,BASE ;CLEAR CONTROL FLAGS
 518 002334  004767  000000G                  JSR     PC,KILL         ;KILL AST (IF THERE WAS ONE)
 519 002340  000167  000000G                  JMP     COMXX
 520                                 ;
 521         000001                          .END
```

```
AHIGH  = ****** GX
ALOW   = ****** GX
ALUCKE = 040000
ALUOE  = 004000
APLACE = ****** GX
A01    = 010000
BASE   = ****** GX
BHIGH  = ****** GX
BINWD  = ****** GX
BITVAL = 000000
BIT0   = 000001
BIT1   = 000002
BIT10  = 002000
BIT11  = 004000
BIT12  = 010000
BIT13  = 020000
BIT14  = 040000
BIT15  = 100000
BIT2   = 000004
BIT3   = 000010
BIT4   = 000020
BIT5   = 000040
BIT6   = 000100
BIT7   = 000200
BIT8   = 000400
BIT9   = 001000
BLOW   = ****** GX
BUFFSA = ****** GX
BUFSET = ****** GX
BUFS2  = ****** GX
BYTE0  = 000000
BYTE1  = 000001
BYTE10 = 000012
BYTE11 = 000013
BYTE12 = 000014
BYTE13 = 000015
BYTE14 = 000016
BYTE15 = 000017
BYTE16 = 000020
BYTE17 = 000021
BYTE18 = 000022
BYTE19 = 000023
BYTE2  = 000002
BYTE20 = 000024
BYTE21 = 000025
BYTE22 = 000026
BYTE23 = 000027
BYTE24 = 000030
BYTE25 = 000031
BYTE26 = 000032
BYTE27 = 000033
BYTE28 = 000034
BYTE29 = 000035
BYTE3  = 000003
BYTE30 = 000036
BYTE31 = 000037
BYTE32 = 000040
BYTE33 = 000041
BYTE34 = 000042
BYTE35 = 000043
BYTE36 = 000044
BYTE37 = 000045
BYTE38 = 000046
BYTE39 = 000047
BYTE4  = 000004
BYTE40 = 000050
BYTE41 = 000051
BYTE42 = 000052
BYTE43 = 000053
BYTE44 = 000054
BYTE45 = 000055
BYTE46 = 000056
BYTE47 = 000057
BYTE48 = 000060
BYTE49 = 000061
BYTE5  = 000005
BYTE50 = 000062
BYTE51 = 000063
BYTE52 = 000064
BYTE53 = 000065
BYTE54 = 000066
BYTE55 = 000067
BYTE56 = 000070
BYTE57 = 000071
BYTE58 = 000072
BYTE59 = 000073
BYTE6  = 000006
BYTE60 = 000074
BYTE61 = 000075
BYTE62 = 000076
BYTE63 = 000077
BYTE64 = 000100
BYTE65 = 000101
BYTE66 = 000102
BYTE67 = 000103
BYTE68 = 000104
BYTE69 = 000105
BYTE7  = 000007
BYTE70 = 000106
BYTE71 = 000107
BYTE72 = 000110
BYTE73 = 000111
BYTE74 = 000112
BYTE75 = 000113
BYTE76 = 000114
BYTE77 = 000115
BYTE78 = 000116
BYTE79 = 000117
BYTE8  = 000010
BYTE80 = 000120
BYTE81 = 000121
BYTE82 = 000122
BYTE83 = 000123
BYTE84 = 000124
BYTE85 = 000125
BYTE86 = 000126
BYTE87 = 000127
BYTE88 = 000130
BYTE89 = 000131
BYTE9  = 000011
BYTE90 = 000132
BYTE91 = 000133
BYTE92 = 000134
BYTE93 = 000135
BYTE94 = 000136
BYTE95 = 000137
BYTE96 = 000140
BYTE97 = 000141
BYTE98 = 000142
BYTE99 = 000143
BYTVAL = 000144
CBKALL = 001000
CBKCLK = 000400
CHIGH  = ****** GX
CLOW   = ****** GX
CNOBRE = 100000
COMXX  = ****** GX
CPCCEN = 010000
CPREAD = 040000
CPWRTE = 020000
CSADRD = 000004
CSEQCI = 100000
CSOE   = 000040
CSWRTE = 000100
DATA1  = ****** GX
DBR.RD = 000001
DB$CPP = 001457
DB$SPT = 000026
DB$TPC = 000023
DISPGS = 100000
DMAAWR = 000005
DMARRD = 000003
DMARWR = 000004
ENBR   = 010000
ERR1   = ****** GX
ERR3   = ****** GX
ERR4   = ****** GX
ERR6   = ****** GX
FIND   = ****** GX
HANG   = ****** GX
IHIGH  = ****** GX
ILOW   = ****** GX
INCVAL = ****** GX
KILL   = ****** GX
LDAM   000332RG
LDBM   000366RG
LDBT   001112RG
LDCM   000422RG
LDCNTL = ****** GX
LDIB   000456RG
LDOB   000654RG
LD1    000060RG
LD1LN  = 000006 G
LD1TBL 000000RG
LD1X   001344R
LOC.EN = 000100
LOC.WA = 040000
LOC.WB = 100000
LOOP   = ****** GX
MAREN1 = 000001
MAREN2 = 004000
MARLOD = 010000
MAROUT = 000002
MAR.LO = 002000
MAR.OU = 000040
MBKALL = 001000
MBKCLK = 000400
MEND   = ****** GX
MMADRD = 000100
MMLEFT = 000002
MMOE   = 000004
MMWRTE = 000010
MNOBRE = 100000
MREN1  = 000001
MREN2  = 020000
MSTRT  = ****** GX
MSTR2  = ****** GX
MSYN   = 000040
N      = 000144
OHIGH  = ****** GX
OLOW   = ****** GX
ONCE   = ****** GX
OUT    = ****** GX
PACK   = ****** GX
PDATA  = ****** GX
PLB    = 000010
PLC    = 000020
PLD    = 000030
PLRWR  = 000200
PLR.EN = 000200
PRAM   001650RG
PRBM   001704RG
PRBT   002210RG
PRCM   001740RG
PRCNTL = ****** GX
PRDATA = ****** GX
PRIB   001774RG
PROB   002072RG
PR1    001362RG
PR1LN  = 000006 G
PR1TBL 000030RG
002 PR1X   002326R
002 QR$CR1 = 176420
002 QR$CR2 = 176422
002 QR$LBR = 176424
    Q$ATTN = 000100
002 Q$BCL  = 000001
002 Q$CCCP = 000040
002 Q$CHB  = 000400
    Q$CHRL = 000200
002 Q$CLR  = 000040
002 Q$CNC  = 030000
    Q$CP   = 000060
    Q$CPCC = 000010
    Q$CP2  = 000260
    Q$CSC  = 010000
    Q$CSEL = 000360
    Q$CSET = 000002
    Q$CSP  = 020000
    Q$DMA  = 000001
    Q$ENBK = 040000
    Q$ENOP = 020000
    Q$FAL  = 004000
    Q$FC   = 000045
    Q$FO   = 000044
    Q$FP   = 000046
    Q$HBF  = 000002
    Q$ICP  = 000006
    Q$IHB  = 000003
    Q$IHRL = 000002
    Q$IMRP = 000007
    Q$LBD  = 001000
    Q$LBDP = 001001
    Q$LBP  = 000001
    Q$LDCD = 000003
    Q$LDMD = 000004
    Q$LDPP = 002000
    Q$LHP  = 010000
    Q$MNC  = 140000
    Q$MR   = 000052
    Q$MRP  = 000040
    Q$MRP2 = 000240
    Q$MSC  = 040000
    Q$MSET = 000004
    Q$MSP  = 100000
    Q$NCLK = 176000
    Q$PP   = 000100
002 Q$PPSW = 000320
002 Q$PP2  = 000300
002 Q$QHLT = 000013
002 Q$QL   = 000043
    Q$QLA  = 000053
    Q$QLB  = 000054
002 Q$QLR  = 000001
002 Q$QW   = 000042
002 Q$RDCD = 000005
    Q$RDMD = 000006
002 Q$REBK = 001000
002 Q$RNC  = 006000
    Q$RSC  = 004000
    Q$RSET = 000010
    Q$SM   = 100000
    Q$SP   = 000120
    Q$SP2  = 000340
    RGQ.EN = 000200
```

```
RGO.VA=·020000          T$CLK· =·002000         WORD1 =·000002·         WORD40=·000120          WORD71=·000216
RP     =·****** GX·     T$DISK=·000200          WORD10=·000024          WORD41=·000122·         WORD72=·000220
RTNPT  =·****** GX·     T$DRD  =·000004         WORD11=·000026          WORD42=·000124          WORD73=·000222
SCAN·  =·****** GX·     T$EMEM=·010000          WORD12=·000030          WORD43=·000126          WORD74=·000224
SEQ.CI=·000010          T$FSAA=·000000          WORD13=·000032·         WORD44=·000130          WORD75=·000226
S$CLR·=·000000          T$FSAB=·000004          WORD14=·000034          WORD45=·000132·         WORD76=·000230
S$LA·  =·000001         T$FSAC=·000014          WORD15=·000036          WORD46=·000134          WORD77=·000232
S$QB·  =·000005         T$FSB2=·000010          WORD16=·000040          WORD47=·000136          WORD78=·000234
S$QR·  =·000006         T$IB·  =·000026         WORD17=·000042·         WORD48=·000140          WORD79=·000236
S$QX·  =·000004         T$IBAR=·000024          WORD18=·000044          WORD49=·000142·         WORD8 =·000020
S$SR·  =·000007         T$IBE· =·020000         WORD19=·000046          WORD5 =·000012·         WORD80=·000240
S$S1   =·000010         T$IBF· =·040000         WORD2· =·000004         WORD50=·000144          WORD81=·000242
S$S2·  =·000014         T$ICD· =·000040         WORD20=·000050          WORD51=·000146          WORD82=·000244
TD$CTR=·176370          T$MODE=·004000          WORD21=·000052·         WORD52=·000150          WORD83=·000246
TD$CTW=·176360          T$OB·  =·000036         WORD22=·000054          WORD53=·000152·         WORD84=·000250
TD$INL=·004000          T$OBE· =·004000         WORD23=·000056          WORD54=·000154          WORD85=·000252
TD$MEM=·000270          T$OBF· =·010000         WORD24=·000060          WORD55=·000156          WORD86=·000254
TD$OAR=·176344·         T$OBRA=·000034          WORD25=·000062·         WORD56=·000160          WORD87=·000256
TD$OTR=·176346          T$OBWA=·000032·         WORD26=·000064          WORD57=·000162·         WORD88=·000260
TD$ORD=·000274          T$OUTA=·100000          WORD27=·000066          WORD58=·000164          WORD89=·000262
TD$SW·=·176376          T$RBDO=·000200          WORD28=·000070          WORD59=·000166          WORD9 =·000022
TD$TAR=·176372·         T$RNB· =·000040         WORD29=·000072·         WORD6 =·000014          WORD90=·000264
TD$TAW=·176362·         T$RSET=·040000          WORD3 =·000006          WORD60=·000170          WORD91=·000266
TD$TDR=·176374          T$SC·  =·000022·        WORD30=·000074          WORD61=·000172·         WORD92=·000270
TD$TDW=·176364          T$SCLK=·020000          WORD31=·000076          WORD62=·000174          WORD93=·000272
THIGH·=·****** GX·      T$SEG1=·000000          WORD32=·000100          WORD63=·000176          WORD94=·000274
TLOW·  =·****** GX·     T$SEG2=·000001          WORD33=·000102·         WORD64=·000200          WORD95=·000276
T$AD·  =·000020         T$SEG3=·000002·         WORD34=·000104          WORD65=·000202·         WORD96=·000300
T$BA·  =·000002·        T$SO·  =·000001         WORD35=·000106          WORD66=·000204          WORD97=·000302
T$BD·  =·000010         T$UBUS=·100000          WORD36=·000110          WORD67=·000206          WORD98=·000304
T$BSO·=·100000          T$1CLK=·000400          WORD37=·000112·         WORD68=·000210          WORD99=·000306
T$BT·  =·000020         T$8BEN=·000020          WORD38=·000114          WORD69=·000212·         WRDVAL=·000310
T$BTAR=·000030          UBD.IN=·000020          WORD39=·000116          WORD7 =·000016          XTREAD=·001000
T$BTD·=·002000          WORD0 =·000000          WORD4 =·000010          WORD70=·000214          XTWRTE=·000400
T$CD·  =·000100

. ABS.  000000     000
        000000     001
TMEM·   002344     002·
ERRORS·DETECTED:  0

VIRTUAL·MEMORY·USED:  3537 WORDS· (·14 PAGES)
DYNAMIC·MEMORY:  4916 WORDS· ( 18 PAGES)
ELAPSED·TIME:  00:00:51
[MEM,TMEM/-SP=[20,1]IM,[20,1]TMEM·
```

```
                                        .TITLE  TDATA
000000                                  .PSECT  TDATA
                                        .LIST   MEB
                                ;
                                ;
                                ;       TERM DETECTOR 'MANUAL' DEBUGGING AIDS
                                ;       DATA COMMANDS (INPUT BUFFER AND BYTE TRANSLATOR)
                                ;
                                ;       COMMANDS:
                                ;       DA      FILL LOCAL BUFFER WITH DATA FOR TD INPUT BUFFER
                                ;       CH      CHANGE DATA IN LOCAL BUFFER OR IN LOCAL BYTE
                                ;               TRANSLATOR TABLES
                                ;       LI      LIST CONTENTS OF LOCAL DATA BUFFER OR LOCAL
                                ;               BYTE TRANSLATOR TABLES
                                ;       EN      MARK THE END OF THE LOCAL DATA BUFFER
                                ;       DF      READ LOCAL DATA BUFFER OR LOCAL BYTE TRANSLATOR
                                ;               TABLES FROM DISK
                                ;       TR      TRANSFER LOCAL DATA BUFFER TO TERM DETECTOR
                                ;               INPUT BUFFER OR LOCAL BYTE TRANSLATOR TO
                                ;               TERM DETECTOR BYTE TRANSLATOR
                                ;
                                ;       ONCE A COMMAND HAS BEEN EXECUTED (OR AN ERROR ENCOUNTERED)
                                ;       THIS MODULE RETURNS CONTROL TO THE MODULE TMAIN AT LOCATION
                                ;       'COMXX'.
                                ;
                                ;
                                        .MCALL  OPEN$R,GET$,CLOSE$
                                ;
                                ;
                                ;       TABLE OF VALID BUFFER MNEMONICS AND THEIR ASSOCIATED
                                ;       ROUTINE ADDRESSES
                                ;
000000                          CH1TBL::
000000  111     102                     .ASCII  /IB/            ;INPUT BUFFER
000002  000260'                         .WORD   CHIB
000004  102     124                     .ASCII  /BT/            ;BYTE TRANSLATOR
000006  000376'                         .WORD   CHBT
        000002                  CH1LN   ==      <.-CH1TBL>/4
                                ;
                                ;       LIST
                                ;
000010                          LI1TBL::
000010  111     102                     .ASCII  /IB/            ;INPUT BUFFER
000012  000672'                         .WORD   LIIB
000014  102     124                     .ASCII  /BT/            ;BYTE TRANSLATOR
000016  001072'                         .WORD   LIBT
        000002                  LI1LN   ==      <.-LI1TBL>/4
                                ;
                                ;       DATA FILES
                                ;
000020                          DF1TBL::
000020  111     102                     .ASCII  /IB/            ;INPUT BUFFER
000022  001366'                         .WORD   DFIB
000024  102     124                     .ASCII  /BT/            ;BYTE TRANSLATOR
000026  001576'                         .WORD   DFBT
        000002                  DF1LN   ==      <.-DF1TBL>/4
                                ;
```

```
                                ;       TRANSFER
                                ;
                                TR1TBL::
000030                                  .ASCII  /IB/            ;INPUT BUFFER
000030      111     102
000032  002134'                         .WORD   TRIB
000034      102     124                 .ASCII  /BT/            ;BYTE TRANSLATOR
000036  002226'                         .WORD   TRBT
        000002                  TR1LN   ==      <.-TR1TBL>/4
                                ;
                                ;
                                ;       DATA
                                ;
                                ;
                                ;       NO OPERANDS
                                ;       THIS ROUTINE PUTS OUT PROMPTS FOR DATA AND READS THE COMMAND
                                ;       LINE, IT THEN CALLS A SUBROUTINE IN TMAIN WHICH PROCESSES
                                ;       THE COMMAND LINE AND PLACES IT INTO TWO BUFFERS, ONE OF THE
                                ;       BUFFERS IS SIMPLY A MIRROR OF THE COMMAND LINE INPUT TO BE
                                ;       USED FOR VERIFICATION, THE OTHER BUFFER IS THE 6-BIT BYTE
                                ;       BUFFER WHICH WILL EVENTUALLY BE TRANSFERRED TO THE
                                ;       TERM DETECTOR INPUT BUFFER (SEE THE 'TR' AND 'RU'
                                ;       COMMANDS).
                                ;
                                ;       NB. SINCE THE LOCAL INPUT BUFFER IS TRANSFERRED TO
                                ;       THE TERM DETECTOR IN 16-BIT WORDS AND THERE ARE 8
                                ;       6-BIT BYTES IN 3 16-BIT WORDS, THE NUMBER OF 6-BIT
                                ;       BYTES IN THE LOCAL DATA BUFFER SHOULD BE DIVISIBLE
                                ;       BY 8. NO CHECKING IS DONE, HOWEVER.
                                ;
                                ;       CLEAR LOCAL DATA BUFFER (BOTH THE 6-BIT BYTE TABLE
                                ;       AND THE 8-BIT ASCII MIRROR TABLE).
                                ;
                                ;
000040                          DA1::
000040  005067  000000G                 CLR     COUNT           ;CLEAR BIT SHIFT COUNTER
000044  005067  000000G                 CLR     DATALN          ;CLEAR LENGTH OF 'DATA'
000050  012700  000000G                 MOV     #DTBL,R0        ;POINT TO 6-BIT BYTE TABLE
000054  012701  001000                  MOV     #<<1364.*3>/8.>+1,R1 ;NUMBER OF WORDS IN TABLE
000060  005020                  1$:     CLR     (R0)+           ;CLEAR DATA TABLE
000062  005301                          DEC     R1
000064  001375                          BNE     1$
                                ;
000066  012700  000000G                 MOV     #DSAVE,R0       ;POINT TO ASCII MIRROR TABLE
000072  012701  001252                  MOV     #1364./2,R1     ;NUMBER OF WORDS
000076  005020                  2$:     CLR     (R0)+           ;CLEAR TABLE
000100  005301                          DEC     R1
000102  003375                          BGT     2$
                                ;
                                ;       READ 'DATA' FROM CONSOLE
                                ;       CALL ROUTINE TO ADD 1 COMMAND LINE'S WORTH OF DATA
                                ;       TO THE TABLES, A <CR> RESPONSE TO THE PROMPT
                                ;       TERMINATES THE COMMAND.
                                ;
000104  012704  000000G                 MOV     #DTBL,R4        ;R4 -> 6-BIT BYTE TABLE
000110  012705  000000G                 MOV     #DSAVE,R5       ;R5 -> MIRROR INPUT TABLE
000114  004767  000000G         DATA:   JSR     PC,GCONLY       ;GET COMMAND LINE
000120  005767  000000G                 TST     GCMLEN          ;ANYTHING THERE
```

```
115 000124  001407                          BEQ     DA1X            ;NO, EXIT
116 000126  004767  000000G                 JSR     PC,DBLD         ;ELSE PROCESS DATA CHARS
117 000132  103404                          BCS     DA1X            ;DATA OVERFLOW OR ERROR
118 000134  066767  000000G 000000G         ADD     DPLUS,DATALN    ;ACCUMULATE LENGTH OF 'DATA'
119 000142  000764                          BR      DATA            ;GET ANOTHER LINE
120                                 ;
121 000144                          DA1X:
122 000144  000167  000000G                 JMP     COMXX
```

```
                                ;
                                ;
                                ;       CHANGE
                                ;       PERFORM SECOND LEVEL PARSING
                                ;       EG. IN THE COMMAND:
                                ;       >CH IB 0
                                ;       PARSE THE 'IB'
                                ;
                                ;
000150                          CH1::
000150  004767  000000G         1$:     JSR     PC,FIND         ;FIND THE BUFFER MNEMONIC IN COMMAND LINE
000154  103004                  2$:     BCC     1$              ;OK, CONTINUE
000156  004767  000000G                 JSR     PC,ERR3         ;'MISSING OPERAND'
000162  000167  000360                  JMP     CH1X
                                ;
                                ;       MATCH THE MNEMONIC FROM THE COMMAND LINE AGAINST THE
                                ;       TABLE OF VALID MNEMONICS.
                                ;
000166  012700  000002          1$:     MOV     #CH1LN,R0       ;NUMBER OF TABLE ENTRIES
000172  012702  000000'                 MOV     #CH1TBL,R2      ;TABLE OF RESPONSES/RTN ADDRESSES
000176  004767  000000G                 JSR     PC,SCAN         ;CHECK COMMAND LINE AGAINST TABLE
000202  103004                          BCC     2$              ;OK, CONTINUE
000204  004767  000000G                 JSR     PC,ERR6         ;'INVALID MEMORY MNEMONIC'
000210  000167  000332                  JMP     CH1X
                                ;
                                ;       SAVE THE POINTER TO THE ROUTINE ASSOCIATED WITH THE
                                ;       BUFFER; R1 -> ROUTINE ADDRESS,
                                ;       SCAN THE COMMAND LINE FOR THE 'START CHANGE ADDRESS'.
                                ;       THE START ADDRESS IS ZERO-RELATIVE.
                                ;
000214  010167  000000G         2$:     MOV     R1,RTNPT        ;SAVE POINTER TO RTN ADDRESS
000220  004767  000000G                 JSR     PC,FIND         ;LOCATE START CHANGE ADDR IN COMMAND LINE
000224  103003                          BCC     3$              ;OK, CONTINUE
000226  004767  000000G                 JSR     PC,ERR3         ;'MISSING OPERAND'
000232  000545                          BR      CH1X            ;EXIT
000234  004767  000000G         3$:     JSR     PC,PACK         ;CONVERT COMMAND LINE VALUE TO BINARY
000240  103003                          BCC     4$              ;CONVERSION SUCCESSFUL
000242  004767  000000G                 JSR     PC,ERR4         ;'INVALID NUMERIC VALUE'
000246  000537                          BR      CH1X
                                ;
                                ;       JUMP TO THE ROUTINE TO CHANGE BUFFER CONTENTS.
                                ;
000250  016701  000000G         4$:     MOV     RTNPT,R1        ;POINT TO ROUTINE
000254  000171  000000                  JMP     @(R1)           ;GO THERE
                                ;
                                ;
                                ;       INPUT BUFFER
                                ;
                                ;       NB. SINCE THE LOCAL INPUT BUFFER IS TRANSFERRED TO
                                ;       THE TERM DETECTOR IN 16-BIT WORDS AND THERE ARE 8
                                ;       6-BIT BYTES IN 3 16-BIT WORDS, THE NUMBER OF 6-BIT
                                ;       BYTES IN THE LOCAL DATA BUFFER SHOULD BE DIVISIBLE
                                ;       BY 8. NO CHECKING IS DONE, HOWEVER.
                                ;
                                ;
000260                          CHIB::
000260  005767  000000G         1$:     TST     DATALN          ;HAS 'DATA' COMMAND BEEN EXECUTED
```

```
181 000264  001003                          BNE·    40$             ;YES, CONTINUE·
182·000266  004767  000000G·                JSR·    PC,ERR7         ;DATA·MUST·PRECEED CHANGE·
183 000272· 000525                          BR      CH1X
184                                 ;
185                                 ;       DETERMINE·WHERE·IN·THE TWO·TABLES·(6-BIT·BYTE·AND·8-BIT·
186                                 ;       ASCII·MIRROR TABLE) TO START·CHANGING. DO·THIS·FOR·THE·
187                                 ;       6-BIT·BYTE·TABLE·BY·CONVERTING·THE·START·CHANGE·ADDRESS·
188                                 ;       FROM·THE·COMMAND·LINE·INTO·A·AN·OFFSET·INTO·THE·6-BIT·
189                                 ;       BYTE·TABLE. THE·FIELD·'OFF6' IS·THE·OUTPUT FROM·THE·
190                                 ;       ROUTINE·OFFSET. THE·FIELD·'BINWD' IS·THE·START·CHANGE·
191                                 ;       ADDRESS·FROM THE·COMMAND·LINE·(IN·BINARY).
192·                                ;
193 000274  004767  000000G·        40$:    JSR·    PC,OFFSET·      ;CALCULATE·6-BIT·TABLE·OFFSET·
194 000300  012704  000000G·                MOV·    #DTBL,R4        ;R4 -> 6-BIT·TABLE
195 000304  066704  000000G·                ADD·    OFF6,R4         ;START·CHANGING·HERE·
196 000310  012705  000000G·                MOV·    #DSAVE,R5       ;R5 -> 8-BIT·MIRROR·TABLE·
197 000314  066705  000000G·                ADD·    BINWD,R5        ;START·CHANGING·HERE·
198 000320  016767  000000G·000000G·        MOV·    BINWD,CHLEN·    ;INIT·FOR·NEW·LENGTH·OF·BUFFER
199                                 ;
200                                 ;       PUT·OUT·PROMPT·FOR·CHANGE·DATA, CALL·SUBROUTINE·TO·
201                                 ;       PROCESS·COMMAND·LINE·(SEE·'DA' COMMAND·ABOVE),
202·                                ;       ADD·THE·NUMBER·OF·6-BIT·CHARACTERS·FROM·THE·
203                                 ;       PROCESSED·COMMAND·LINE TO·THE·CHANGE·START
204                                 ;       ADDRESS.
205                                 ;
206 000326  004767  000000G·        5$:     JSR·    PC,GCONLY·      ;GET·COMMAND·LINE·
207 000332· 005767  000000G·                TST·    GCMLEN·         ;ANYTHING·THERE·
208 000336  001407                          BEQ·    6$              ;NO, <CR> RESPONSE
209 000340  004767  000000G·                JSR·    PC,DBLD·        ;CHANGE·'DATA' TABLE·
210 000344  103500                          BCS·    CH1X            ;ERROR, EXIT·
211 000346  066767  000000G·000000G·        ADD·    DPLUS,CHLEN·    ;ACCUMULATE·LENGTH OF·DATA·BUFFER·
212·000354  000764                          BR      5$              ;PROMPT·FOR·MORE·CHANGE·DATA
213                                 ;
214                                 ;       SPECIAL·PROCESSING:
215                                 ;       THE·FIELD·'DATALN' CONTAINS·THE·NUMBER·OF·6-BIT·BYTES·
216                                 ;       IN THE·LOCAL DATA·BUFFER·(SEE·'DA' COMMAND). IF·THE·
217                                 ;       CHANGE·COMMAND·IS·IN·THE·FORM:
218                                 ;       >CH·1B·2·
219                                 ;       AND·THE·DATA BUFFER·CONTAINS·'ABCDE' AND·THE·CHANGE·DATA·IS·
220                                 ;       '1234', THE·DATA·BUFFER·WILL·CONTAIN·'AB1234' AND·DATALN·WILL·
221                                 ;       BE UPDATED·TO·6. IF, HOWEVER, THE·CHANGE·DATA·IS·'12' THEN·
222·                                ;       THE·DATA·BUFFER·WILL·CONTAIN·'AB12D' AND·DATALN·WILL REMAIN·
223                                 ;       AT 5. THE·FIELD·'DATALN' CONTAINS·THE·LENGTH·OF·THE·DATA·
224                                 ;       BUFFER·BEFORE·ANY·CHANGES. THE·FIELD·'CHLEN' CONTAINS·
225                                 ;       THE·LENGTH·OF·THE·BUFFER·AFTER·THE·CHANGES. THE·LARGER·
226                                 ;       OF THE·TWO·WILL·BECOME THE·NEW·'DATALN'.
227                                 ;
228·000356  026767  000000G·000000G·6$:    CMP·    CHLEN,DATALN·   ;WHICH·IS·BIGGER·
229 000364  003470                          BLE·    CH1X            ;DATALN·
230 000366  016767  000000G·000000G·        MOV·    CHLEN,DATALN·   ;REPLACE DATALN·
231 000374  000464                          BR      CH1X            ;EXIT·
232·                                ;
233                                 ;
234                                 ;       BYTE·TRANSLATOR·
235                                 ;       THE·FIELD·'BINWD' CONTAINS·THE·CHANGE·START·ADDRESS,
236                                 ;       WHICH·IS·A·ZERO-RELATIVE·WORD·NUMBER, EG. THE·COMMAND:
237                                 ;       >CH·BT·7
```

```
                                        ;       WILL·START·CHANGING·THE·BYTE·TRANSLATOR·OVERRIDE·TABLE·
                                        ;       (SEE·BELOW) AT·WORD·7.
                                        ;
                                        ;
000376                                  CHBT::
000376  026727  000000G·000000C·        CMP·    BINWD,#<BTEND-BTRANS> ;START·ADDRESS·IN·RANGE·
000404  103403                          BLO·    3$              ;YES, CONTINUE·
000406  004767  000000G·                JSR·    PC,ERR20        ;'START·ADDRESS OUT·OF·RANGE HIGH·
000412·000455                           BR      CH1X            ;EXIT·
                                        ;
                                        ;       THE·CHANGE·COMMAND·CREATES·A·BYTE·TRANSLATOR·OVERRIDE·TABLE·
                                        ;       (WHICH·IS·A·COMPLETE·COPY·OF·THE BYTE·TRANSLATOR·TABLE). ALL·
                                        ;       ALL·CHANGES·ARE·APPLIED·TO·THE·OVERRIDE·TABLE. THE·OVERRIDE·
                                        ;       TABLE·IS·TRANSFERRED·TO·THE·TERM DETECTOR·BY·EITHER·THE·'TR'
                                        ;       OR THE·'RU' COMMAND. IF·THIS·ROUTINE·IS·BEING·ENTERED·FOR·
                                        ;       THE·FIRST·TIME, COPY·THE·BYTE·TRANSLATOR·TABLE·TO·THE·
                                        ;       OVERRIDE·TABLE.
                                        ;
000414  032767  000000G·000000G·3$:      BIT·    #RIDE,BASE·     ;FIRST·USE·OF·'CHANGE'
000422  001014                          BNE·    4$              ;NO, LEAVE·OVERRIDE·TABLE·ALONE·
000424  052767  000000G·000000G·        BIS·    #RIDE,BASE·     ;FLAG·ROUTINE·ENTERED·
000432·012700  000000G·                 MOV·    #BTRANS,R0      ;POINT·TO·DEFAULT·TABLE·
000436  012701  000000G·                MOV·    #BTOVER,R1      ;POINT·TO·OVERRIDE TABLE·
000442·012702·000000C·                  MOV·    #<BTEND-BTRANS>/2,R2 ;NUMBER·OF·WORDS·
000446  012021                  30$:    MOV·    (R0)+,(R1)+     ;LOAD·OVERRIDE·TABLE·
000450  005302·                        DEC·    R2·
000452·001375                           BNE·    30$
                                        ;
                                        ;       LOAD·THE·CHANGE·START·ADDRESS·IN BYTES·INTO·R2·
                                        ;
000454  012702·000000G·          4$:    MOV·    #BTOVER,R2·     ;POINT·TO·OVERRIDE TABLE·
000460  016703  000000G·                MOV·    BINWD,R3        ;LOAD·WORD·OFFSET·
000464  006303                          ASL·    R3              ;SHIFT·FOR·BYTE·OFFSET·
000466  060302·                        ADD·    R3,R2·          ;START·CHANGING·HERE·
                                        ;
                                        ;       PROMPT·FOR·A LINE·OF·CHANGE·DATA. THE·COMMAND·LINE·
                                        ;       MAY·CONTAIN·MORE·THEN·ONE·WORD·OF·DATA, EG:
                                        ;       >000000 000000 000000 000000 000000 000000
                                        ;       CHANGE·6 CONTIGUOUS·WORDS·WORTH·OF·BYTE·TRANSLATOR·CODES.
                                        ;       A·<CR> RESPONSE·TO·THE PROMPT·TERMINATES·THE·COMMAND.
                                        ;
000470  004767  000000G·          5$:    JSR·    PC,GCONLY·      ;READ·COMMAND·LINE
000474  004767  000000G·                JSR·    PC,FIND·        ;LOCATE·AN·OVERRIDE·WORD·
000500  103422·                        BCS·    CH1X            ;<CR> RESPONSE, EXIT·
                                        ;
                                        ;       CONVERT·THE·OVERRIDE·VALUE·TO·BINARY. ANY·ERROR·IN·
                                        ;       CONVERSION·TERMINATES·THE·COMMAND.
                                        ;
000502·004767  000000G·          6$:    JSR·    PC,PACK·        ;CONVERT NUMERIC·VALUE·
000506  103003                          BCC·    8$              ;OK, CONTINUE·
000510  004767  000000G·                JSR·    PC,ERR4         ;'INVALID·NUMERIC·VALUE'
000514  000414                          BR      CH1X            ;EXIT·
                                        ;
                                        ;       CHECK·FOR·TABLE·OVERFLOW. IF·OVERFLOW, TERMINATE·COMMAND.
                                        ;
000516·020227  000000C·          8$:    CMP·    R2,#BTOVER+<BTEND-BTRANS> ;END·OF·TABLE·REACHED·
000522·103403                           BLO·    10$             ;NO, CONTINUE·
```

```
295 000524  004767  000000G         JSR     PC,ERR17        ;TABLE OVERFLOW
296 000530  000406                  BR      CH1X            ;EXIT
297 000532  016722  000000G   10$:   MOV     BINWD,(R2)+     ;ENTER WORD IN BT TABLE
298 000536  004767  000000G         JSR     PC,FIND         ;LOCATE NEXT OVERRIDE WORD
299 000542  103357                  BCC     6$              ;PROCESS STRING
300 000544  000751                  BR      5$              ;GET COMMAND LINE
301                              ;
302                              ;
303 000546               CH1X:
304 000546  000167  000000G         JMP     COMXX
```

```
                                        ;
                                        ;
                                        ;       LIST.
                                        ;       PERFORM SECOND LEVEL PARSING
                                        ;       EG. IN THE COMMAND:
                                        ;       >LI IB 0
                                        ;       PARSE THE 'IB'
                                        ;
                                        ;
000552                                  LI1::
000552  004767  000000G                         JSR     PC,FIND         ;LOCATE BUFFER MNEMONIC IN COMMAND LINE
000556  103004                                  BCC     1$              ;OK, CONTINUE
000560  004767  000000G                         JSR     PC,ERR3         ;'MISSING OPERAND'
000564  000167  000424                         JMP     LI1X            ;EXIT
                                        ;
                                        ;       MATCH THE MNEMONIC FROM THE COMMAND LINE AGAINST THE
                                        ;       TABLE OF VALID MNEMONICS.
                                        ;
000570  012700  000002          1$:     MOV     #LI1LN,R0       ;NUMBER OF TABLE ENTRIES
000574  012702  000010'                 MOV     #LI1TBL,R2      ;POINT TO TABLE
000600  004767  000000G                 JSR     PC,SCAN         ;MATCH AGAINST COMMAND LINE
000604  103004                          BCC     2$              ;MATCH WAS MADE
000606  004767  000000G                 JSR     PC,ERR6         ;'INVALID MEMORY MNEMONIC'
000612  000167  000376                 JMP     LI1X
                                        ;
                                        ;       SAVE THE POINTER TO THE ROUTINE ASSOCIATED WITH THE
                                        ;       BUFFER, R1 -> ROUTINE ADDRESS.
                                        ;       SCAN THE COMMAND LINE FOR THE 'START LIST ADDRESS'.
                                        ;       THE START ADDRESS IS ZERO-RELATIVE.
                                        ;
000616  010167  000000G          2$:     MOV     R1,RTNPT        ;SAVE POINTER
000622  004767  000000G                 JSR     PC,FIND         ;LOCATE LIST START ADDR IN COMMAND LINE
000626  103004                          BCC     3$              ;OK, CONTINUE
000630  004767  000000G                 JSR     PC,ERR3         ;'MISSING OPERAND'
000634  000167  000354                 JMP     LI1X            ;EXIT
000640  004767  000000G          3$:     JSR     PC,PACK         ;CONVERT COMMAND LINE VALUE TO BINARY
000644  103004                          BCC     4$              ;CONVERSION SUCCESSFUL
000646  004767  000000G                 JSR     PC,ERR4         ;'INVALID NUMERIC VALUE'
000652  000167  000336                 JMP     LI1X
                                        ;
                                        ;       JUMP TO THE ROUTINE TO LIST BUFFER CONTENTS.
                                        ;
000656  004767  000000G          4$:     JSR     PC,HANG         ;HOW TO STOP PRINT
000662  016701  000000G                 MOV     RTNPT,R1        ;POINT TO ROUTINE
000666  000171  000000                  JMP     @(R1)           ;JUMP TO ROUTINE
                                        ;
                                        ;
                                        ;       INPUT BUFFER
                                        ;
                                        ;
                                        ;       CHECK WHETHER THERE IS ANY 6-BIT BYTE DATA IN THE
                                        ;       TABLE.
                                        ;
000672                                  LIIB::
000672  005767  000000G                 TST     DATALN          ;HAS 'DATA' COMMAND BEEN EXECUTED
000676  001003                          BNE     20$             ;YES, CONTINUE
000700  004767  000000G                 JSR     PC,ERR8         ;DATA COMMAND MUST PRECEED 'LIST'
```

```
                                                BR      LI1X

                                        ;
                                        ;       CHECK WHETHER THE START LIST ADDRESS IS IN RANGE
                                        ;
000706  026767  000000G 000000G 20$:    CMP     BINWD,DATALN    ;IS LIST START ADDRESS IN RANGE
000714  002403                          BLT     25$             ;YES, CONTINUE
000716  004767  000000G                 JSR     PC,ERR9         ;OUT OF RANGE
000722  000534                          BR      LI1X
                                        ;
                                        ;       LIST FROM THE 8-BIT ASCII MIRROR TABLE, NOT FROM THE
                                        ;       6-BIT BYTE TABLE, SEE THE 'DA' ROUTINE FOR NOTES ON
                                        ;       THE CREATION OF THE MIRROR TABLE. THE FIELD 'BINWD'
                                        ;       CONTAINS THE LIST START ADDRESS.
                                        ;
000724  012700  000000G         25$:    MOV     #DSAVE,R0       ;POINT TO ASCII VERSION OF 'DATA'
000730  066700  000000G                 ADD     BINWD,R0        ;START LIST HERE
000734  005001                          CLR     R1              ;CLEAR PRINT LINE CHAR COUNTER
000736  012705  000000G                 MOV     #PRINT,R5       ;R5 -> PRINT LINE
                                        ;
                                        ;       THE LOOP FLAG IS USED TO CONTROL THE LIST COMMAND,
                                        ;       ANY CHARACTER ENTERED FROM THE TERMINAL ENDS THE
                                        ;       COMMAND AS DOES A 0 VALUE IN THE DATA TABLE.
                                        ;
000742  032767  000000G 000000G 3$:     BIT     #LOOP,BASE      ;FINISHED ?
000750  001521                          BEQ     LI1X            ;YES, EXIT
000752  105710                          TSTB    (R0)            ;END OF TABLE
000754  001003                          BNE     30$             ;NO, CONTINUE
000756  004767  000000G                 JSR     PC,CONSOL       ;PRINT LAST LINE
000762  000514                          BR      LI1X            ;AND EXIT
                                        ;
                                        ;       THE BINARY VALUES 15 (CARRAIGE RETURN) AND 12
                                        ;       (LINE FEED) ARE PRINTED OUT AS <15> AND <12>.
                                        ;
000764  122710  000015          30$:    CMPB    #15,(R0)        ;<CR> IN ASCII DATA
000770  002006                          BGE     4$              ;PROCESS <CR> OF <LF>
000772  005201                          INC     R1              ;ADD TO PRINT LINE CHAR COUNT
000774  022701  000110                  CMP     #72.,R1         ;PRINT LINE FULL
001000  002426                          BLT     7$              ;YES, WRITE TO TT0
001002  112025                          MOVB    (R0)+,(R5)+     ;MOVE CHAR TO PRINT LINE
001004  000756                          BR      3$              ;GET NEXT CHAR
                                        ;
001006  062701  000004          4$:     ADD     #4,R1           ;ADD FOR ANGLE BRACKETS
001012  022701  000110                  CMP     #72.,R1         ;ENOUGH SPACE ON CURRENT LINE
001016  002417                          BLT     7$              ;NO, WRITE CURRENT LINE
001020  112725  000074                  MOVB    #'<,(R5)+       ;MOVE LEFT BRACKET IN ANY CASE
001024  112725  000061                  MOVB    #'1,(R5)+       ;MOVE ASCII 1 IN ANY CASE
001030  122720  000012                  CMPB    #12,(R0)+       ;LINE FEED
001034  001403                          BEQ     5$              ;YES, BUILD ASCII STRING
001036  112725  000065                  MOVB    #'5,(R5)+
001042  000402                          BR      6$              ;FINISH WITH RIGHT BRACKET
001044  112725  000062          5$:     MOVB    #'2,(R5)+
001050  112725  000076          6$:     MOVB    #'>,(R5)+       ;CLOSE BRACKETS
001054  000732                          BR      3$              ;GET NEXT CHAR
                                        ;
                                        ;       WRITE ONE LINE TO THE CONSOLE
                                        ;
001056  004767  000000G         7$:     JSR     PC,CONSOL       ;WRITE LINE TO TT0
```

```
 420 001062  005001                        CLR     R1              ;CLEAR PRINT LINE CHAR COUNT
 421 001064  012705  000000G               MOV     #PRINT,R5       ;RESET PRINT LINE POINTER
 422 001070  000724                        BR      3$              ;PRINT NEXT 78 CHARS
 423                                ;
 424                                ;
 425                                ;       BYTE TRANSLATOR
 426                                ;
 427                                ;
 428                                ;       CHECK WHETHER THE 'CH' COMMAND HAS BEEN USED TO
 429                                ;       CREATE A BYTE TRANSLATOR OVERRIDE TABLE. LIST
 430                                ;       CANNOT BE DONE OTHERWISE.
 431                                ;
 432 001072                         LIBT::
 433 001072  032767  000000G 000000G       BIT     #RIDE,BASE      ;HAS BT TABLE BEEN OVERRIDEN
 434 001100  001003                        BNE     1$              ;YES, CONTINUE
 435 001102  004767  000000G               JSR     PC,ERR22        ;'FOR BT, CHANGE COMMAND MUST PRECEED LIST'
 436 001106  000442                        BR      LI1X            ;EXIT
 437                                ;
 438                                ;       CHECK WHETHER THE START LIST ADDRESS IS IN RANGE.
 439                                ;       THE FIELD 'BINWD' CONTAINS THE WORD OFFSET WHICH
 440                                ;       IS THE START ADDRESS.
 441                                ;       WILL BE USED TO GET THE START ADDRESS.
 442                                ;
 443 001110  026727  000000G 000000C 1$:     CMP     BINWD,#<BTEND-BTRANS> ;IS START ADDRESS IN RANGE
 444 001116  103403                        BLO     2$              ;YES, CONTINUE
 445 001120  004767  000000G               JSR     PC,ERR20        ;'START ADDRESS OUT OF RANGE HIGH'
 446 001124  000433                        BR      LI1X            ;EXIT
 447                                ;
 448                                ;       LIST THE OVERRIDE TABLE.
 449                                ;
 450 001126  012702  000000G       2$:     MOV     #BTOVER,R2      ;POINT TO BT OVERRIDE TABLE
 451 001132  016703  000000G               MOV     BINWD,R3        ;LOAD WORD OFFSET
 452 001136  006303                        ASL     R3              ;SHIFT FOR BYTE OFFSET
 453 001140  060302                        ADD     R3,R2           ;START LIST HERE
 454                                ;
 455                                ;       USE THE LOOP FLAG TO CONTROL PRINTING.
 456                                ;       ANY CHARACTER ENTERED FROM THE CONSOLE TERMINATES
 457                                ;       THE COMMAND AS DOES END OF DATA.
 458                                ;
 459 001142  012705  000000G       3$:     MOV     #PRINT,R5       ;POINT TO PRINT LINE
 460 001146  012703  000005                MOV     #5,R3           ;COUNT NUMBER OF WORDS IN PRINT LINE
 461 001152  012201                4$:     MOV     (R2)+,R1        ;LOAD WORD FROM BT OVERRIDE TABLE
 462 001154  004767  000000G               JSR     PC,UNPK         ;CONVERT FOR PRINTING
 463 001160  020227  000000C               CMP     R2,#BTOVER+<BTEND-BTRANS> ;END OF TABLE REACHED
 464 001164  103011                        BHIS    5$              ;YES, EXIT
 465 001166  005303                        DEC     R3              ;SUB FROM PRINT LINE COUNT
 466 001170  001370                        BNE     4$              ;CONVERT NEXT TABLE ENTRY
 467 001172  004767  000000G               JSR     PC,CONSOL       ;PRINT LINE
 468 001176  032767  000000G 000000G       BIT     #LOOP,BASE      ;LOOP ON
 469 001204  001403                        BEQ     LI1X            ;NO, EXIT
 470 001206  000755                        BR      3$              ;START WITH FRESH LINE
 471                                ;
 472 001210  004767  000000G       5$:     JSR     PC,CONSOL       ;PRINT LAST LINE
 473                                ;
 474                                ;
 475 001214  004767  000000G       LI1X:   JSR     PC,KILL         ;KILL AST (IF THERE WAS ONE)
 476 001220  000167  000000G               JMP     COMXX
```

```
                                        ;
                                        ;
                                        ;       END DATA
                                        ;
                                        ;
                                        ;       CHANGE THE NUMBER OF 6-BIT BYTE CHARACTERS TO BE
                                        ;       TRANSFERRED TO THE TERM DETECTOR INPUT BUFFER. THE
                                        ;       CONTENTS OF THE BUFFER REMAINED UNCHANGED.
                                        ;
                                        ;       NB. SINCE THE LOCAL INPUT BUFFER IS TRANSFERRED TO
                                        ;       THE TERM DETECTOR IN 16-BIT WORDS AND THERE ARE 8
                                        ;       6-BIT BYTES IN 3 16-BIT WORDS, THE NUMBER OF 6-BIT
                                        ;       BYTES IN THE LOCAL DATA BUFFER SHOULD BE DIVISIBLE
                                        ;       BY 8. NO CHECKING IS DONE, HOWEVER.
                                        ;
                                        ;
001224                                  EN1::
001224  004767  000000G                         JSR     PC,FIND         ;LOCATE END VALUE IN COMMAND LINE
001230  103003                                  BCC     3$              ;OK, CONTINUE
001232  004767  000000G                         JSR     PC,ERR3         ;'MISSING OPERAND'
001236  000425                                  BR      EN1X            ;EXIT
001240  004767  000000G                 3$:     JSR     PC,PACK         ;CONVERT COMMAND LINE VALUE TO BINARY
001244  103003                                  BCC     4$              ;CONVERSION SUCCESSFUL
001246  004767  000000G                         JSR     PC,ERR4         ;'INVALID NUMERIC VALUE'
001252  000417                                  BR      EN1X
                                        ;
                                        ;       THE END VALUE CANNOT BE ZERO.
                                        ;
001254  005767  000000G                 4$:     TST     BINWD           ;IS END VALUE ZERO
001260  001003                                  BNE     5$              ;NO, CONTINUE
001262  004767  000000G                         JSR     PC,ERR4         ;REPORT INVALID VALUE
001266  000411                                  BR      EN1X            ;AND EXIT
                                        ;
                                        ;       THE 'DATA' COMMAND MUST HAVE BEEN PREVIOUSLY EXECUTED.
                                        ;       IF IT HAS BEEN, REPLACE THE CONTENTS OF THE FIELD
                                        ;       'DATALN' (LENGTH OF LOCAL DATA BUFFER) WITH THE
                                        ;       END VALUE FROM THE COMMAND LINE.
                                        ;
001270  005767  000000G                 5$:     TST     DATALN          ;HAS 'DATA' COMMAND BEEN EXECUTED
001274  001003                                  BNE     6$              ;YES, CONTINUE
001276  004767  000000G                         JSR     PC,ERR10        ;DATA COMMAND MUST PRECEED 'END'
001302  000403                                  BR      EN1X
001304  016767  000000G 000000G         6$:     MOV     BINWD,DATALN    ;REPLACE LENGTH OF 'DATA'
                                        ;
001312                                  EN1X:
001312  000167  000000G                         JMP     COMXX
```

```
                                ;
                                ;
                                ;       DATA FILES ROUTINES
                                ;
                                ;       READ EITHER THE 'DATA' BUFFER OR THE BYTE TRANSLATOR
                                ;       BUFFER FROM DISK,
                                ;       PERFORM SECOND LEVEL PARSING
                                ;       EG, IN THE COMMAND:
                                ;       >DF IB
                                ;       PARSE THE 'IB'
                                ;
                                ;
001316                          DF1::
001316  004767  000000G         JSR     PC,FIND         ;FIND A BUFFER MNEMONIC IN COMMAND LINE
001322  103004                 BCC     1$              ;OK, CONTINUE
001324  004767  000000G         JSR     PC,ERR3         ;'MISSING OPERAND'
001330  000167  000514          JMP     DF1X
                                ;
                                ;       MATCH THE MNEMONIC FROM THE COMMAND LINE AGAINST THE
                                ;       TABLE OF VALID MNEMONICS,
                                ;
001334  012700  000002          1$:     MOV     #DF1LN,R0       ;NUMBER OF TABLE ENTRIES
001340  012702  000020'                 MOV     #DF1TBL,R2      ;TABLE OF RESPONSES/RTN ADDRESSES
001344  004767  000000G                 JSR     PC,SCAN         ;CHECK COMMAND LINE AGAINST TABLE
001350  103004                         BCC     2$              ;OK, CONTINUE
001352  004767  000000G                 JSR     PC,ERR11        ;'INVALID FILE MNEMONIC'
001356  000167  000466                 JMP     DF1X
001362  000171  000000          2$:     JMP     @(R1)           ;JUMP TO ROUTINE
                                ;
                                ;
                                ;       LOAD 'DATA' BUFFER FROM DISK
                                ;       THIS ROUTINE READS A 'DATA' FILE FROM DISKONE RECORD AT A
                                ;       TIME, IT THEN CALLS A SUBROUTINE IN TMAIN WHICH PROCESSES
                                ;       THE LINE AND PLACES IT INTO TWO BUFFERS, ONE OF THE BUFFERS
                                ;       IS SIMPLY A MIRROR OF THE INPUT LINE INPUT TO BE USED FOR
                                ;       BUFFER VERIFICATION, THE OTHER BUFFER IS THE 6-BIT BYTE
                                ;       BUFFER WHICH WILL EVENTUALLY BE TRANSFERRED TO THE
                                ;       TERM DETECTOR INPUT BUFFER (SEE THE 'TR' AND 'RU'
                                ;       COMMANDS),
                                ;
                                ;       CLEAR LOCAL DATA BUFFER (BOTH THE 6-BIT BYTE TABLE
                                ;       AND THE 8-BIT ASCII MIRROR TABLE),
                                ;
001366                          DFIB::
001366  005067  000000G         CLR     COUNT           ;CLEAR BIT SHIFT COUNTER
001372  005067  000000G         CLR     DATALN          ;CLEAR LENGTH OF 'DATA'
001376  012700  000000G         MOV     #DTBL,R0        ;POINT TO 6-BIT BYTE TABLE
001402  012701  001000          MOV     #<<1364.*3>/8.>+1,R1 ;NUMBER OF WORDS IN TABLE
001406  005020                  1$:     CLR     (R0)+           ;CLEAR DATA TABLE
001410  005301                          DEC     R1
001412  001375                          BNE     1$
                                ;
001414  012700  000000G                 MOV     #DSAVE,R0       ;POINT TO ASCII MIRROR TABLE
001420  012701  001252                  MOV     #1364./2,R1     ;NUMBER OF WORDS
001424  005020                  2$:     CLR     (R0)+           ;CLEAR TABLE
001426  005301                          DEC     R1
001430  003375                          BGT     2$
```

```
                                        ;
                                        ;       OPEN THE FILE 'DATA.TXT'
                                        ;
001432  012704  000000G                 MOV     #DTBL,R4        ;R4 -> 6-BIT BYTE TABLE
001436  012705  000000G                 MOV     #DSAVE,R5       ;R5 -> 8-BIT ASCII MIRROR TABLE
001442                                  OPEN$R  #TXTFDB
001442  012700  000000G                 MOV     #TXTFDB,R0
001446  112760  000000G 000000G         .IIF    NB,#FO.RD,      MOVB #FO.RD,F.FACC(R0)
001454  004767  000000G                 JSR     PC,.OPEN
001460  103005                          BCC     4$
001462  004767  000000G                 JSR     PC,ERR12        ;ERROR ON OPEN
001466  005067  000000G                 CLR     DATALN          ;AS THOUGH RTN WAS NOT ENTERED
001472  000566                          BR      DF1X
                                        ;
                                        ;       READ ONE RECORD. CALL ROUTINE IN TMAIN TO ADD ONE
                                        ;       RECORD'S WORTH OF DATA TO THE LOCAL TABLE.
                                        ;
001474                                  4$:     GET$    #TXTFDB         ;READ 'DATA' FILE
001474  012700  000000G                 MOV     #TXTFDB,R0
001500  004767  000000G                 JSR     PC,.GET
001504  103414                          BCS     5$              ;ERROR OR END OF FILE
001506  016067  000000G 000000G         MOV     F.NRBD(R0),GCMLEN ;LENGTH OF LINE READ
001514  012701  000000G                 MOV     #GCMBUF,R1      ;POINT TO LINE
001520  004767  000000G                 JSR     PC,DBLD         ;BUILD 6-BIT BYTE TABLE
001524  103551                          BCS     DF1X            ;ERROR OR END OF TABLE
001526  066767  000000G 000000G         ADD     DPLUS,DATALN    ;ACCUMULATE LENGTH
001534  000757                          BR      4$              ;GET NEXT LINE
                                        ;
001536  122760  177766  000000G 5$:     CMPB    #-10.,F.ERR(R0) ;END OF FILE
001544  001405                          BEQ     6$              ;YES, PUT OUT MESSAGE
001546  004767  000000G                 JSR     PC,ERR13        ;ERROR ON GET
001552  005067  000000G                 CLR     DATALN          ;AS THOUGH RTN WAS NOT ENTERED
001556  000402                          BR      7$              ;CLOSE FILE AND EXIT
                                        ;
001560  004767  000000G         6$:     JSR     PC,ENFILE
001564                          7$:     CLOSE$  #TXTFDB
001564  012700  000000G                 MOV     #TXTFDB,R0
001570  004767  000000G                 JSR     PC,.CLOSE
001574  000525                          BR      DF1X
                                        ;
                                        ;
                                        ;       LOAD BYTE TRANSLATOR BUFFER FROM DISK.
                                        ;       READ THE FILE 'TRANS.TXT' INTO THE BYTE TRANSLATOR
                                        ;       OVERRIDE TABLE. THE FILE ON DISK MUST BE A FULL 64
                                        ;       WORD TABLE.
                                        ;
                                        ;
001576                          DFBT::
001576  012700  000000G                 MOV     #BTOVER,R0      ;POINT TO BYTE TRANS TABLE
001602  012701  000100                  MOV     #64.,R1         ;NUMBER OF WORDS
001606  005020                  1$:     CLR     (R0)+           ;CLEAR TABLE
001610  005301                          DEC     R1
001612  003375                          BGT     1$
                                        ;
                                        ;       OPEN 'TRANS.TXT'
                                        ;
001614                                  OPEN$R  #TRFDB
```

```
                001614  012700  000000G·                MOV·    #TRFDB,R0
                001620  112760  000000G·000000G·         .IIF·   NB,#FO.RD,      MOVB·#FO.RD,F.FACC(R0)
                001626  004767  000000G·                JSR·    PC,.OPEN·
     632·001632·103003                                  BCC·    2$
     633 001634 004767  000000G·                        JSR·    PC,ERR14        ;ERROR·ON·OPEN·
     634 001640 000503                                  BR      DF1X
     635                                        ;
     636                                        ;       TOP·OF·LOOP, READ·FILE.
     637                                        ;
     638 001642· 012702· 000000G·               2$:     MOV·    #BTOVER,R2·     ;R2·-> BYTE·TRANS TABLE·
     639 001646                                 3$:     GET$    #TRFDB·
                001646  012700  000000G·                MOV·    #TRFDB,R0
                001652· 004767  000000G·                JSR·    PC,.GET·
     640 001656 103021                                  BCC·    6$              ;OK, PROCESS·RECORD·
     641 001660 122760  177766  000000G·                CMPB·   #-10.,F.ERR(R0) ;END·OF·FILE·
     642·001666 001403                                  BEQ·    20$             ;YES, EXIT·
     643 001670 004767  000000G·                        JSR·    PC,ERR15        ;ERROR·ON·GET·
     644 001674 000447                                  BR      30$             ;CLOSE·FILE·
     645                                        ;
     646                                        ;       EXIT·ON·SUCCESSFUL·FILLING·OF·TABLE,
     647                                        ;
     648 001676  052767  000000G·000000G·20$:           BIS·    #RIDE,BASE·     ;SET·FLAG·FOR·OVERRIDE·TABLE FILLED·
     649 001704  004767  000000G·                       JSR·    PC,ENFILE·      ;PRINT·END-OF-FILE MESSAGE·
     650 001710                                         CLOSE$  #TRFDB·
                001710  012700  000000G·                MOV·    #TRFDB,R0
                001714  004767  000000G·                JSR·    PC,.CLOSE·
     651 001720  000453                                 BR      DF1X            ;EXIT·
     652·                                       ;
     653                                        ;       PROCESS·ONE·RECORD, EACH·RECORD·MAY·CONTAIN·SEVERAL·TABLE·
     654                                        ;       VALUES·SEPARATED·BY·BLANKS.  IN·ORDER·TO·PARSE·THEM EASILY
     655                                        ;       MAKING·USE·OF·EXISTING SUBROUTINES, THE·RECORDS·ARE·READ·
     656                                        ;       INTO·THE·COMMAND·LINE·BUFFER·GCMBUF.
     657                                        ;
     658 001722· 016067  000000G·000000G·6$:            MOV·    F.NRBD(R0),GCMLEN·;PRETEND·DISK·RECORD·IS·A·COMMAND·LINE·
     659 001730  012701  000000G·                       MOV·    #GCMBUF,R1      ;POINT·TO·RECORD·READ·
     660 001734  004767  000000G·                       JSR·    PC,FIND·        ;LOCATE·A·TABLE·VALUE·
     661 001740  103003                                 BCC·    7$              ;OK, CONTINUE·
     662·001742· 004767  000000G·                       JSR·    PC,ERR16        ;EMPTY·INPUT·LINE·
     663 001746  000422·                                BR      30$             ;CLOSE·FILE·AND·EXIT·
     664                                        ;
     665 001750  004767  000000G·               7$:     JSR·    PC,PACK·        ;CONVERT·NUMERIC·VALUE·FROM·DISK·RECORD·
     666 001754  103003                                 BCC·    8$              ;OK, CONTINUE·
     667 001756  004767  000000G·                       JSR·    PC,ERR4         ;'INVALID·NUMERIC·VALUE'
     668 001762· 000414                                 BR      30$             ;CLOSE·FILE·AND·EXIT·
     669                                        ;
     670 001764  020227  000000C·               8$:     CMP·    R2,#BTOVER+<BTEND-BTRANS> ;END·OF·TABLE·REACHED·
     671 001770  103403                                 BLO·    10$             ;NO, CONTINUE·
     672·001772· 004767  000000G·                       JSR·    PC,ERR17        ;TABLE·OVERFLOW
     673 001776  000406                                 BR      30$             ;CLOSE·FILE·AND·EXIT·
     674 002000  016722· 000000G·               10$:    MOV·    BINWD,(R2)+     ;ENTER·WORD IN BT TABLE·
     675 002004  004767  000000G·                       JSR·    PC,FIND·        ;LOCATE·NEXT·TABLE VALUE·
     676 002010  103357                                 BCC·    7$              ;PROCESS STRING·
     677 002012· 000715                                 BR      3$              ;NOTHING THERE; FINISHED·WITH·THIS·RECORD.
     678                                        ;
     679                                        ;       EXIT·ON·ERROR, CLEAR·OVERRIDE·TABLE·AND·FLAG.
     680                                        ;
     681 002014                                 30$:    CLOSE$  #TRFDB·
```

```
002014  012700  000000G          MOV     #TRFDB,R0
002020  004767  000000G          JSR     PC,.CLOSE
682                         ;
683 002024  042767  000000G 000000G    BIC     #RIDE,BASE      ;CLEAR BT OVERRIDE FLAG
684 002032  012700  000000G          MOV     #BTOVER,R0      ;POINT TO OVERRIDE TABLE
685 002036  012701  000400           MOV     #256.,R1        ;NUMBER OF WORDS IN TABLE
686 002042  005020              35$:    CLR     (R0)+           ;CLEAR TABLE
687 002044  005301                      DEC     R1
688 002046  001375                      BNE     35$
689                         ;
690                         ;
691 002050                  DF1X:
692 002050  000167  000000G          JMP     COMXX
```

```
                                ;
                                ;
                                ;       TRANSFER
                                ;       MOVE LOCAL DATA TABLE TO TERM DETECTOR INPUT BUFFER
                                ;       OR BYTE TRANSLATOR OVERRIDE TABLE TO TERM DETECTOR
                                ;       BYTE TRANSLATOR.
                                ;       PERFORM SECOND LEVEL PARSING
                                ;       EG. IN THE COMMAND:
                                ;       >TR IB
                                ;       PARSE THE 'IB'
                                ;
                                ;
002054                          TR1::
002054  004767  000000G         1$:     JSR     PC,FIND         ;FIND THE BUFFER MNEMONIC IN COMMAND LINE
002060  103003                          BCC     1$              ;OK, CONTINUE
002062  004767  000000G                 JSR     PC,ERR3         ;'MISSING OPERAND'
002066  000532                          BR      TR1X
                                ;
                                ;       MATCH THE MNEMONIC IN THE COMMAND LINE AGAINST THE
                                ;       TABLE OF VALID MNEMONICS.
                                ;
002070  012700  000002          1$:     MOV     #TR1LN,R0       ;NUMBER OF TABLE ENTRIES
002074  012702  000030'                 MOV     #TR1TBL,R2      ;TALE OF RESPONSES/RTN ADDRESSES
002100  004767  000000G                 JSR     PC,SCAN         ;MATCH COMMAND LINE AGAINST TABLE
002104  103003                          BCC     2$              ;OK, CONTINUE
002106  004767  000000G                 JSR     PC,ERR18        ;'INVALID TABLE NAME'
002112  000520                          BR      TR1X
                                ;
                                ;       SAVE THE POINTER TO THE ROUTINE ASSOCIATED WITH THE
                                ;       BUFFER. R1 -> ROUTINE ADDRESS.
                                ;       CALL A ROUTINE TO SCAN THE COMMAND LINE FOR A LOOP
                                ;       INDICATOR. EG:
                                ;       >TR IB L
                                ;       LOOP FLAG WILL BE SET IF INDICATOR IS PRESENT.
                                ;
002114  010167  000000G         2$:     MOV     R1,RTNPT        ;SAVE POINTER TO ROUTINE
002120  004767  000000G                 JSR     PC,LOOPR        ;LOOP ON TRANSFER
002124  016701  000000G         TRIN:   MOV     RTNPT,R1        ;POINT TO RTN
002130  000171  000000                  JMP     @(R1)           ;JUMP TO ROUTINE
                                ;
                                ;
                                ;       TRANSFER FROM LOCAL DATA BUFFER TO TERM DETECTOR
                                ;       INPUT BUFFER.  THE FIELD 'DATALN' CONTAINS THE NUMBER
                                ;       OF 6-BIT BYTE CHARACTERS IN THE LOCAL BUFFER. CONVERT
                                ;       THIS VALUE INTO A NUMBER PDP 11 WORDS FOR CONTROLLING
                                ;       THE TRANSFER.
                                ;
                                ;
002134                          TRIB::
002134  012767  000000  176376          MOV     #0,TD$SW        ;RESET TD
002142  012767  000200  176360          MOV     #T$DISK,TD$CTW  ;SIMULATE DISK INPUT
002150  016701  000000G                 MOV     DATALN,R1       ;NUMBER OF 6-BIT BYTES
002154  006301                          ASL     R1              ;MULT BY 2
002156  016702  000000G                 MOV     DATALN,R2       ;NUMBER OF 6-BIT BYTES
002162  060201                          ADD     R2,R1           ;RESULT = NUMBER OF BYTES X 3
002164  006201                          ASR     R1              ;NOW DIVIDE BY 8.
002166  006201                          ASR     R1              ;FOR NUMBER
```

```
 751 002170  006201                                ASR·     R1              ;OF·WORDS·
 752·002172· 016767  000000G·000000G·               MOV·     ILOW,MSTR2·     ;IB·START·ADDRESS·
 753 002200  012702· 000000G·                       MOV·     #DTBL,R2·       ;R2·-> 6-BIT·BYTE·TABLE·
 754                                        ;
 755                                        ;       TRANSFER·THE·TABLE·
 756                                        ;
 757 002204  016703  000000G·           1$:      MOV·     MSTR2,R3        ;POINT TO·IB
 758 002210  012213                                MOV·     (R2)+,(R3)      ;LOAD·IB
 759 002212· 062767  000002  000000G·              ADD·     #2,MSTR2·       ;ADVANCE ADDRESS·
 760 002220  005301                                DEC·     R1
 761 002222· 003370                                BGT·     1$
 762·002224  000447                                BR       TRLP            ;TEST·LOOP·FLAG·
 763                                        ;
 764                                        ;
 765                                        ;       TRANSFER·BYTE·TRANSLATOR·TABLE·TO·TD·
 766                                        ;
 767                                        ;
 768                                        ;       IF A·BYTE·TRANSLATOR·OVERRIDE·TABLE·HAS·BEEN·BUILT
 769                                        ;       (SEE·THE·'CH' AND·'DF' COMMANDS), THEN·TRANSFER·IT
 770                                        ;       TO THE·TERM·DETECTOR. OTHERWISE·TRANSFER·THE·DEFAULT
 771                                        ;       BYTE·TRANSLATOR·TABLE. IN·EITHER CASE·TRANSFER·THE·
 772·                                       ;       SAME·TABLE·FOUR·TIMES.
 773                                        ;
 774 002226                                 TRBT::
 775 002226  012767  000000  176376                MOV·     #0,TD$SW·       ;RESET·TD·
 776 002234  012702· 000004·                       MOV·     #4,R2·          ;TRANSFER·SAME·TABLE·4 TIMES
 777 002240  016767  000000G·000000G·              MOV·     TLOW,MSTR2·     ;LOAD·START·ADDRESS·
 778 002246  012700  000100·            1$:      MOV·     #64.,R0         ;NUMBER·OF·WORDS·IN·TABLE·
 779 002252· 032767  000000G·000000G·              BIT      #RIDE,BASE·     ;IS·OVERRIDE·TABLE IN·USE·
 780 002260  001403                                BEQ·     2$              ;NO·
 781 002262· 012701  000000G·                      MOV·     #BTOVER,R1      ;POINT TO·OVERRIDE TABLE·
 782·002266  000402·                               BR       3$
 783 002270  012701  000000G·           2$:      MOV·     #BTRANS,R1      ;R1 -> BYTE·TRANS TABLE·
 784 002274  012767  000040  176360     3$:      MOV·     #T$ICD,TD$CTW·  ;SET·INTERFACE·AND CONTROL·
 785 002302· 012767  000030· 176362·    4$:      MOV·     #T$BTAR,TD$TAW  ;SELECT·BYTE·TRANS MAR·
 786 002310  016767  000000G·176364               MOV·     MSTR2,TD$TDW
 787 002316  012767  000020  176362·              MOV·     #T$BT,TD$TAW·   ;SELECT·BYTE·TRANS MEMORY·
 788 002324  012167  176364                        MOV·     (R1)+,TD$TDW·
 789 002330  005267  000000G·                      INC·     MSTR2·          ;ADVANCE ADDRESS·
 790 002334  005300                                DEC·     R0
 791 002336  003361                                BGT·     4$
 792·002340  005302·                               DEC      R2·             ;SUB·FROM·OUTER·LOOP·COUNT·
 793 002342· 001341                                BNE·     1$
 794                                        ;
 795 002344                                 TRLP:
 796 002344  032767  000000G·000000G·              BIT·     #LOOP,BASE·     ;LOOP·ON TRANSFER·
 797 002352· 001264                                BNE·     TRIN            ;YES, REPEAT·
 798                                        ;
 799 002354                                 TR1X:
 800 002354  004767  000000G·                      JSR·     PC,KILL·        ;KILL·AN AST·(IF·THERE·WAS·ONE)
 801 002360  000167  000000G·                      JMP·     COMXX·
 802·                                       ;
 803         000001                                .END·
```

```
ALUCKE= 040000
ALUOE = 004000
A01   = 010000
BASE  = ****** GX
BINWD = ****** GX
BITVAL= 000000
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BTEND = ****** GX
BTOVER= ****** GX
BTRANS= ****** GX
BYTE0 = 000000
BYTE1 = 000001
BYTE10= 000012
BYTE11= 000013
BYTE12= 000014
BYTE13= 000015
BYTE14= 000016
BYTE15= 000017
BYTE16= 000020
BYTE17= 000021
BYTE18= 000022
BYTE19= 000023
BYTE2 = 000002
BYTE20= 000024
BYTE21= 000025
BYTE22= 000026
BYTE23= 000027
BYTE24= 000030
BYTE25= 000031
BYTE26= 000032
BYTE27= 000033
BYTE28= 000034
BYTE29= 000035
BYTE3 = 000003
BYTE30= 000036
BYTE31= 000037
BYTE32= 000040
BYTE33= 000041
BYTE34= 000042
BYTE35= 000043
BYTE36= 000044
BYTE37= 000045
BYTE38= 000046
BYTE39= 000047
BYTE4 = 000004
BYTE40= 000050
BYTE41= 000051
BYTE42= 000052
BYTE43= 000053
BYTE44= 000054
BYTE45= 000055
BYTE46= 000056
BYTE47= 000057
BYTE48= 000060
BYTE49= 000061
BYTE5 = 000005
BYTE50= 000062
BYTE51= 000063
BYTE52= 000064
BYTE53= 000065
BYTE54= 000066
BYTE55= 000067
BYTE56= 000070
BYTE57= 000071
BYTE58= 000072
BYTE59= 000073
BYTE6 = 000006
BYTE60= 000074
BYTE61= 000075
BYTE62= 000076
BYTE63= 000077
BYTE64= 000100
BYTE65= 000101
BYTE66= 000102
BYTE67= 000103
BYTE68= 000104
BYTE69= 000105
BYTE7 = 000007
BYTE70= 000106
BYTE71= 000107
BYTE72= 000110
BYTE73= 000111
BYTE74= 000112
BYTE75= 000113
BYTE76= 000114
BYTE77= 000115
BYTE78= 000116
BYTE79= 000117
BYTE8 = 000010
BYTE80= 000120
BYTE81= 000121
BYTE82= 000122
BYTE83= 000123
BYTE84= 000124
BYTE85= 000125
BYTE86= 000126
BYTE87= 000127
BYTE88= 000130
BYTE89= 000131
BYTE9 = 000011
BYTE90= 000132
BYTE91= 000133
BYTE92= 000134
BYTE93= 000135
BYTE94= 000136
BYTE95= 000137
BYTE96= 000140
BYTE97= 000141
BYTE98= 000142
BYTE99= 000143
BYTVAL= 000144
CBKALL= 001000
CBKCLK= 000400
CHBT    000376RG
CHIB    000260RG
CHLEN = ****** GX
CH1     000150RG
CH1LN = 000002 G
CH1TBL  000000RG
CH1X    000546R
CNOBRE= 100000
COMXX = ****** GX
CONSOL= ****** GX
COUNT = ****** GX
CPCCEN= 010000
CPREAD= 040000
CPWRTE= 020000
CSADRD= 000004
CSEQCI= 100000
CSOE  = 000040
CSWRTE= 000100
DATA    000114R
DATALN= ****** GX
DA1     000040RG
DA1X    000144R
DBLD  = ****** GX
DBR.RD= 000001
DB$CPP= 001457
DB$SPT= 000026
DB$TPC= 000023
DFBT    001576RG
DFIB    001366RG
DFI     001316RG
DFILN = 000002 G
DFITBL  000020RG
DFIX    002050R
DISPGS= 100000
DMAAWR= 000005
DMARRD= 000003
DMARWR= 000004
DPLUS = ****** GX
DSAVE = ****** GX
DTBL  = ****** GX
ENBR  = 010000
ENFILE= ****** GX
EN1     001224RG
EN1X    001312R
ERR10 = ****** GX
ERR11 = ****** GX
ERR12 = ****** GX
ERR13 = ****** GX
ERR14 = ****** GX
ERR15 = ****** GX
ERR16 = ****** GX
ERR17 = ****** GX
ERR18 = ****** GX
ERR20 = ****** GX
ERR22 = ****** GX
ERR3  = ****** GX
ERR4  = ****** GX
002 ERR6  = ****** GX
002 ERR7  = ****** GX
ERR8  = ****** GX
002 ERR9  = ****** GX
FIND  = ****** GX
002 FO.RD = ****** GX
002 F.ERR = ****** GX
F.FACC= ****** GX
F.NRBD= ****** GX
GCMBUF= ****** GX
GCMLEN= ****** GX
GCONLY= ****** GX
HANG  = ****** GX
ILOW  = ****** GX
KILL  = ****** GX
LIBT    001072RG
LIIB    000672RG
LII     000552RG
002 LI1LN = 000002 G
LI1TBL  000010RG
002 LI1X    001214R
002 LOC.EN= 000100
LOC.WA= 040000
LOC.WB= 100000
LOOP  = ****** GX
LOOPR = ****** GX
MAREN1= 000001
002 MAREN2= 004000
002 MARLOD= 010000
002 MAROUT= 000002
MAR.LO= 002000
002 MAR.OU= 000040
002 MBKALL= 001000
MBKCLK= 000400
MMADRD= 000100
MMLEFT= 000002
MMOE  = 000004
MMWRTE= 000010
MNOBRE= 100000
MREN1 = 000001
MREN2 = 020000
MSTR2 = ****** GX
002 MSYN  = 000040
002 N     = 000144
OFFSET= ****** GX
OFF6  = ****** GX
PACK  = ****** GX
PAR$$$= 000027
PLB   = 000010
PLC   = 000020
PLD   = 000030
PLRWR = 000200
PLR.EN= 000200
PRINT = ****** GX
QR$CR1= 176420
QR$CR2= 176422
QR$LBR= 176424
Q$ATTN= 000100
Q$BCL = 000001
Q$CCCP= 000040
Q$CHB = 000400
Q$CHRL= 000200
Q$CLR = 000040
Q$CNC = 030000
Q$CP  = 000060
Q$CPCC= 000010
Q$CP2 = 000260
Q$CSC = 010000
Q$CSEL= 000360
Q$CSET= 000002
Q$CSP = 020000
Q$DMA = 000001
002 Q$ENBK= 040000
002 Q$ENOP= 020000
002 Q$FAL = 004000
Q$FC  = 000045
002 Q$FO  = 000044
002 Q$FP  = 000046
Q$HBF = 000002
Q$ICP = 000006
Q$IHB = 000003
Q$IHRL= 000002
Q$IMRP= 000007
Q$LBD = 001000
Q$LBDP= 001001
Q$LBP = 000001
Q$LDCD= 000003
Q$LDMD= 000004
Q$LDPP= 002000
Q$LHP = 010000
Q$MNC = 140000
Q$MR  = 000052
Q$MRP = 000040
Q$MRP2= 000240
Q$MSC = 040000
Q$MSET= 000004
Q$MSP = 100000
Q$NCLK= 176000
Q$PP  = 000100
Q$PPSW= 000320
```

SYMBOL TABLE

```
Q$PP2 =  000300          TD$TDW=  176364          T$RBDO=  000200          WORD32=  000100          WORD69=  000212
Q$QHLT=  000013          TLOW  =  ****** GX       T$RNB =  000040          WORD33=  000102          WORD7 =  000016
Q$QL  =  000043          TRBT     002226RG    002 T$RSET=  040000          WORD34=  000104          WORD70=  000214
Q$QLA =  000053          TRFDB =  ****** GX       T$SC  =  000022          WORD35=  000106          WORD71=  000216
Q$QLB =  000054          TRIB     002134RG    002 T$SCLK=  020000          WORD36=  000110          WORD72=  000220
Q$QLR =  000001          TRIN     002124R     002 T$SEG1=  000000          WORD37=  000112          WORD73=  000222
Q$QW  =  000042          TRLP     002344R     002 T$SEG2=  000001          WORD38=  000114          WORD74=  000224
Q$RDCD=  000005          TR1      002054RG    002 T$SEG3=  000002          WORD39=  000116          WORD75=  000226
Q$RDMD=  000006          TR1LN =  000002 G        T$SO  =  000001          WORD4 =  000010          WORD76=  000230
Q$REBK=  001000          TR1TBL   000030RG    002 T$UBUS=  100000          WORD40=  000120          WORD77=  000232
Q$RNC =  006000          TR1X     002354R     002 T$1CLK=  000400          WORD41=  000122          WORD78=  000234
Q$RSC =  004000          TXTFDB=  ****** GX       T$8BEN=  000020          WORD42=  000124          WORD79=  000236
Q$RSET=  000010          T$AD  =  000020          UBD.IN=  000020          WORD43=  000126          WORD8 =  000020
Q$SM  =  100000          T$BA  =  000002          UNPK  =  ****** GX       WORD44=  000130          WORD80=  000240
Q$SP  =  000120          T$BD  =  000010          WORD0 =  000000          WORD45=  000132          WORD81=  000242
Q$SP2 =  000340          T$BSO =  100000          WORD1 =  000002          WORD46=  000134          WORD82=  000244
RGQ.EN=  000200          T$BT  =  000020          WORD10=  000024          WORD47=  000136          WORD83=  000246
RGQ.VA=  020000          T$BTAR=  000030          WORD11=  000026          WORD48=  000140          WORD84=  000250
RIDE  =  ****** GX       T$BTD =  002000          WORD12=  000030          WORD49=  000142          WORD85=  000252
RTNPT =  ****** GX       T$CD  =  000100          WORD13=  000032          WORD5 =  000012          WORD86=  000254
SCAN  =  ****** GX       T$CLK =  002000          WORD14=  000034          WORD50=  000144          WORD87=  000256
SEQ.CI=  000010          T$DISK=  000200          WORD15=  000036          WORD51=  000146          WORD88=  000260
S$CLR =  000000          T$DRD =  000004          WORD16=  000040          WORD52=  000150          WORD89=  000262
S$LA  =  000001          T$EMEM=  010000          WORD17=  000042          WORD53=  000152          WORD9 =  000022
S$QB  =  000005          T$FSAA=  000000          WORD18=  000044          WORD54=  000154          WORD90=  000264
S$QR  =  000006          T$FSAB=  000004          WORD19=  000046          WORD55=  000156          WORD91=  000266
S$QX  =  000004          T$FSAC=  000014          WORD2 =  000004          WORD56=  000160          WORD92=  000270
S$SR  =  000007          T$FSB2=  000010          WORD20=  000050          WORD57=  000162          WORD93=  000272
S$S1  =  000010          T$IB  =  000026          WORD21=  000052          WORD58=  000164          WORD94=  000274
S$S2  =  000014          T$IBAR=  000024          WORD22=  000054          WORD59=  000166          WORD95=  000276
TD$CTR=  176370          T$IBE =  020000          WORD23=  000056          WORD6 =  000014          WORD96=  000300
TD$CTW=  176360          T$IBF =  040000          WORD24=  000060          WORD60=  000170          WORD97=  000302
TD$INL=  004000          T$ICD =  000040          WORD25=  000062          WORD61=  000172          WORD98=  000304
TD$MEM=  000270          T$MODE=  004000          WORD26=  000064          WORD62=  000174          WORD99=  000306
TD$OAR=  176344          T$OB  =  000036          WORD27=  000066          WORD63=  000176          WRDVAL=  000310
TD$OTR=  176346          T$OBE =  004000          WORD28=  000070          WORD64=  000200          XTREAD=  001000
TD$ORD=  000274          T$OBF =  010000          WORD29=  000072          WORD65=  000202          XTWRTE=  000400
TD$SW =  176376          T$OBRA=  000034          WORD3 =  000006          WORD66=  000204          .CLOSE=  ****** G
TD$TAR=  176372          T$OBWA=  000032          WORD30=  000074          WORD67=  000206          .GET  =  ****** G
TD$TAW=  176362          T$OUTA=  100000          WORD31=  000076          WORD68=  000210          .OPEN =  ****** G
TD$TDR=  176374
```

```
. ABS.  000000     000
        000000     001
TDATA   002364     002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  4595 WORDS  ( 18 PAGES)
DYNAMIC MEMORY:  5972 WORDS  ( 22 PAGES)
ELAPSED TIME:  00:01:01
!DATA,TDATA/-SP=[20,1]JIM,[20,1]TDATA
```

```
                                        .TITLE  TRUN
                                        .PSECT  TRUN
                                        .LIST   MEB
                                ;
                                ;
                                ;       TERM DETECTOR 'MANUAL' DEBUGGING AIDS
                                ;       RUN COMMANDS
                                ;
                                ;       COMMANDS:
                                ;       RU      RUN TERM DETECTOR
                                ;
                                ;       ONCE A COMMAND HAS BEEN EXECUTED (OR AN ERROR ENCOUNTERED)
                                ;       THIS MODULE RETURNS CONTROL TO THE MODULE TMAIN,AT LOCATION
                                ;       'COMXX'.
                                ;
                                ;
                                RU1TBL::
000000     123     103          .ASCII  /SC/            ;SINGLE CLOCK
000002  000070'                 .WORD   RUSC
000004     121     120          .ASCII  /QP/            ;READ OUTPUT TO QR AND PRINT
000006  000124'                 .WORD   RUQP
000010     121     116          .ASCII  /QN/            ;READ OUTPUT TO QR (NO PRINT)
000012  000134'                 .WORD   RUQN
000014     116     103          .ASCII  /NC/            ;NORMAL CONTINUOUS CLOCK
000016  000144'                 .WORD   RUNC
        000004          RU1LN   ==      <.-RU1TBL>/4
                                ;
                                ;
                                ;       RUN
                                ;       PERFORM SECOND LEVEL PARSING
                                ;       EG. IN THE COMMAND:
                                ;       >RU NC
                                ;       PARSE THE 'NC'
                                ;
000020                          RU1::
000020  004767  000000G         JSR     PC,FIND         ;FIND AN OPERAND IN COMMAND LINE
000024  103004                  BCC     1$              ;OK, CONTINUE
000026  004767  000000G         JSR     PC,ERR3         ;'MISSING OPERAND'
000032  000167  000532          JMP     RU1X
                                ;
                                ;       MATCH THE OPERAND FROM THE COMMAND LINE AGAINST A TABLE
                                ;       OF VALID OPERANDS
                                ;
000036  012700  000004  1$:     MOV     #RU1LN,R0       ;NUMBER OF TABLE ENTRIES
000042  012702  000000'         MOV     #RU1TBL,R2      ;TABLE OF RESPONSES/RTN ADDRESSES
000046  004767  000000G         JSR     PC,SCAN         ;MATCH COMMAND LINE AGAINST TABLE
000052  103004                  BCC     2$              ;OK, CONTINUE
000054  004767  000000G         JSR     PC,ERR18        ;'INVALID TABLE NAME'
000060  000167  000504          JMP     RU1X
                                ;
                                ;       R1 -> ADDRESS OF ROUTINE TO BE EXECUTED.
                                ;
000064  000171  000000  2$:     JMP     @(R1)           ;JUMP TO ROUTINE
                                ;
                                ;
                                ;       RUN WITH A SINGLE CLOCK
                                ;
```

```
                                ;       $ALL·A·ROUTINE·TO·SCAN·THE·COMMAND·LINE·FOR·A·
                                ;       LOOP·INDICATOR. EG:
                                ;       >RU·SC·L·
                                ;       LOOP·FLAG·WILL·BE·SET·IF·INDICATOR·IS·PRESENT.
                                ;
                                ;
000070                          RUSC::
000070  004767  000000G·                JSR·    PC,LOOPR·       ;LOOP·ON CLOCK·?·
000074  012767  000000  176376          MOV·    #0,TD$SW·       ;RESET·TD·
000102· 012767  024000  176360  1$:     MOV·    #<T$SCLK+T$MODE>,TD$CTW·
000110  032767  000000G·000000G·        BIT·    #LOOP,BASE·     ;LOOP·FLAG·ON·?·
000116  001371                          BNE·    1$              ;YES, REPEAT·
000120  000167  000444                  JMP·    RU1X
                                ;
                                ;
                                ;       RUN·NORMAL·
                                ;       READ·TD·OUTPUT·TO·QR·AND·PRINT·
                                ;
                                ;
000124                          RUQP::
000124  052767  000000G·000000G·        BIS·    #QP,BASE·       ;SET·FLAG·FOR·PRINT·
000132· 000404                          BR      RUNC            ;RUN·NORMAL·
                                ;
                                ;
                                ;       RUN·NORMAL·
                                ;       READ·TD·OUTPUT·TO·QR·(NO·PRINT)
                                ;
                                ;
000134                          RUQN::
000134  052767  000000G·000000G·        BIS·    #QN,BASE·       ;SET·FLAG·FOR·NO·PRINT·
000142· 000400                          BR      RUNC
                                ;
                                ;
                                ;       RUN·NORMAL·
                                ;       ISSUE·QIO·FOR·UNSOLICITED·INTERRUPT·FROM·TERMINAL·
                                ;       (THE·LOOP·FLAG·WILL·BE SET). THE LOOP·FLAG WILL·
                                ;       ALLOW·CONTINUOUS·RUNNING·OF·THE·TERM·DETECTOR·UNTIL·
                                ;       A·CHARACTER·IS·ENTERED FROM·THE·TERMINAL.
                                ;
                                ;       LOAD·BYTE·TRANSLATOR. IF·A·BYTE·TRANSLATOR OVERRIDE·
                                ;       TBALE·HAS·BEEN·BUILT·(SEE·THE·'CH' AND·'DF' COMMANDS)
                                ;       THEN·TRANSFER·IT·TO·THE·TERM·DETECTOR. OTHERWISE·TRANSFER·
                                ;       THE·DEFAULT·BYTE·TRANSLATOR·TABLE.
                                ;
                                ;
000144                          RUNC::
000144  004767  000000G·                JSR·    PC,HANG·        ;ALLOW·FOR·STOPPING·TEST·
000150  012767  000000  176376          MOV·    #0,TD$SW·       ;RESET·TD·
000156  012702· 000004                  MOV·    #4,R2·          ;TRANSFER·SAME·TABLE·4 TIMES
000162· 016767  000000G·000000G·        MOV·    TLOW,MSTR2·     ;LOAD·START·ADDRESS·
000170  012700  000100          100$:   MOV·    #64.,R0         ;NUMBER·OF·WORDS·IN·TABLE·
000174  032767  000000G·000000G·        BIT·    #RIDE,BASE·     ;IS·OVERRIDE·TABLE IN·USE·
000202· 001403                          BEQ·    12$             ;NO·
000204  012701  000000G·                MOV·    #BTOVER,R1      ;POINT·TO·OVERRIDE TABLE·
000210  000402·                        BR      14$
000212· 012701  000000G·        12$:    MOV·    #BTRANS,R1      ;R1 -> BYTE·TRANS·TABLE·
000216  012767  000040  176360  14$:    MOV·    #T$ICD,TD$CTW·  ;SET·INTERFACE AND CONTROL
```

```
115 000224  012767  000030  176362  10$:    MOV     #T$BTAR,TD$TAW  ;SELECT BYTE TRANS MAR
116 000232  016767  000000G 176364          MOV     MSTR2,TD$TDW
117 000240  012767  000020  176362          MOV     #T$BT,TD$TAW    ;SELECT BYTE TRANS MEMORY
118 000246  012167  176364                 MOV     (R1)+,TD$TDW
119 000252  005267  000000G                INC     MSTR2           ;ADVANCE ADDRESS
120 000256  005300                         DEC     R0
121 000260  003361                         BGT     10$
122 000262  005302                         DEC     R2              ;SUB FROM OUTER LOOP COUNT
123 000264  001341                         BNE     100$
124                                 ;
125                                 ;       TRANSFER FROM THE LOCAL DATA BUFFER TO THE TERM DETECTOR
126                                 ;       INPUT BUFFER. THE FIELD 'DATALN' CONTAINS THE NUMBER OF
127                                 ;       6-BIT BYTE CHARACTERS IN THE LOCAL BUFFER. CONVERT THIS
128                                 ;       VALUE INTO A NUMBER OF PDP 11 WORDS FOR CONTROLLING THE
129                                 ;       TRANSFER. 4 6-BIT BYTES FIT INTO 3 8-BIT BYTES, 8 6-BIT
130                                 ;       BYTES FIT INTO 3 16-BIT WORDS.
131                                 ;
132 000266  016701  000000G                MOV     DATALN,R1       ;NUMBER OF 6-BIT BYTES
133 000272  006301                         ASL     R1              ;MULT BY 2
134 000274  016702  000000G                MOV     DATALN,R2       ;NUMBER OF 6-BIT BYTES
135 000300  060201                         ADD     R2,R1           ;RESULT = NUMBER OF BYTES X 3
136 000302  006201                         ASR     R1              ;NOW DIVIDE BY 8.
137 000304  006201                         ASR     R1              ;FOR NUMBER OF
138 000306  006201                         ASR     R1              ;WORDS
139 000310  010146                         MOV     R1,-(SP)        ;SAVE FOR REPEATED USE
140                                 ;
141                                 ;       FILL UP THE TERM DETECTOR INPUT BUFFER. MOVE THE
142                                 ;       LOCAL DATA BUFFER TO THE TERM DETECTOR REPEATEDLY
143                                 ;       UNTIL THE TD INPUT BUFFER IS FULL. WRAP AROUND ON BOTH
144                                 ;       BUFFERS (IE. SINCE THE AMOUNT OF DATA IN THE LOCAL DATA
145                                 ;       BUFFER WILL PROBABLY NOT BE ENOUGH TO FILL THE TD INPUT
146                                 ;       BUFFER, TRANSFER THE SAME DATA REPEATEDLY).
147                                 ;
148 000312  011601                         MOV     (SP),R1         ;NUMBER OF WORDS IN 'DATA' TABLE
149 000314  012702  000000G                MOV     #DTBL,R2        ;R2 -> DATA TABLE
150 000320  016703  000000G                MOV     ILOW,R3         ;R3 -> INPUT BUFFER
151 000324  012767  000200  176360         MOV     #T$DISK,TD$CTW  ;SIMULATE DISK INPUT
152 000332  012223                 1$:     MOV     (R2)+,(R3)+     ;MOVE FROM PGM BUFFER TO TD INPUT BUFFER
153 000334  016704  176376                 MOV     TD$SW,R4        ;LOAD TD STATUS REG
154 000340  032704  040000                 BIT     #T$IBF,R4       ;TEST FOR INPUT BUFFER FULL
155 000344  001011                         BNE     2$              ;FULL, START CLOCK NOW
156 000346  020367  000000G                CMP     R3,IHIGH        ;RUN OUT OF MEMORY
157 000352  003006                         BGT     2$              ;YES, START CLOCK
158 000354  005301                         DEC     R1              ;SUB FROM NUMBBER OF WORDS
159 000356  003365                         BGT     1$              ;MOVE NEXT
160 000360  011601                         MOV     (SP),R1         ;RELOAD NUMBER OF WORDS IN 'DATA' TABLE
161 000362  012702  000000G                MOV     #DTBL,R2        ;R2 -> 'DATA' TABLE
162 000366  000761                         BR      1$              ;KEEP MOVING
163                                 ;
164                                 ;       RUN THE TERM DETECTOR. WHEN THE TD INPUT BUFFER GOES
165                                 ;       'NOT FULL' TRANSFER FROM THE LOCAL DATA BUFFER TO THE TD
166                                 ;       INPUT BUFFER . R2 -> LOCAL DATA BUFFER, R3 -> TD INPUT BUFFER.
167                                 ;
168 000370  012767  024200  176360  2$:     MOV     #<T$DISK+T$SCLK+T$MODE>,TD$CTW  ;SIMULATE DISK INPUT
169 000376  032767  000000G 000000G 3$:     BIT     #LOOP,BASE      ;LOOP?
170 000404  001470                         BEQ     RUST            ;NO, RESTORE SP AND EXIT
171 000406  016704  176376                 MOV     TD$SW,R4        ;LOAD TD STATUS REG
```

```
000412· 032704  040000                  BIT·    #T$IBF,R4       ;IS·INPUT·BUFFER·FULL·
000416  001367                          BNE·    3$              ;YES, WAIT·FOR·NOT FULL·
000420  020367  000000G·                 CMP·    R3,IHIGH·       ;RUN·OUT OF·INPUT·BUFFER·
000424  003402·                         BLE·    4$              ;NO, CONTINUE·
000426  016703  000000G·                 MOV·    ILOW,R3         ;RE-INIT INPUT·BUFFER·POINTER·
000432· 005301                  4$:     DEC·    R1              ;FINISHED·WITH·'DATA' BUFFER
000434  003003                          BGT·    5$              ;NO, CONTINUE·
000436  011601                          MOV·    (SP),R1         ;RELOAD·NUMBER·OF·WORDS·IN·'DATA'
000440  012702· 000000G·                 MOV·    #DTBL,R2·       ;RE-INIT 'DATA' TABLE·POINTER·
000444  012223                  5$:     MOV·    (R2)+,(R3)+     ;LOAD·INPUT·BUFFER
                                ;
                                ;       READ·TD·OUTPUT·AVAILABLE·REGISTER, ACT·DEPENDING·ON·
                                ;       PRINT·OPTIONS·SET·ABOVE:
                                ;
                                ;       READ·WITHOUT PRINT:
                                ;
000446  032767  000000G·000000G·         BIT·    #QN,BASE·       ;READ·TD OUTPUT·AND·NO·PRINT
000454  001410                          BEQ·    7$              ;NO, TRY READ·AND·PRINT·
000456  016704  176344          6$:     MOV·    TD$OAR,R4       ;READ·OUTPUT·AVAILABLE·REGISTER·
000462· 032704  100000                  BIT·    #T$OUTA,R4      ;IS·OUTPUT·AVAILABLE·
000466· 001743                          BEQ·    3$              ;NO, CONTINUE·RUN·
000470  016705  176346                  MOV·    TD$OTR,R5       ;READ·OUTPUT·
000474  000770                          BR      6$              ;READ·UNTIL·EMPTY·
                                ;
                                ;       READ·AND·PRINT·
                                ;
000476  032767  000000G·000000G· 7$:     BIT·    #QP,BASE·       ;READ·OUTPUT·AND·PRINT·
000504  001734                          BEQ·    3$              ;NO, CONTINUE·RUN·
000506  016704  176344          8$:     MOV·    TD$OAR,R4       ;READ·OUTPUT·AVAILABLE·REGISTER·
000512· 032704  100000                  BIT·    #T$OUTA,R4      ;IS·OUTPUT·AVAILABLE·
000516· 001006                          BNE·    9$              ;YES, PRINT·IT·
000520  012767  046505  000000G·         MOV·    #"EM,PRINT·     ;EMPTY·
000526  004767  000000G·                 JSR·    PC,CONSOL·      ;PRINT·MESSAGE·
000532· 000721                          BR      3$              ;TRY·AGAIN·
000534  032767  000000G·000000G· 9$:     BIT·    #LOOP,BASE·     ;LOOP·ON ?·
000542· 001411                          BEQ·    RUST            ;NO, EXIT·
000544  012705  000000G·                 MOV·    #PRINT,R5       ;POINT·TO·PRINT·LINE·
000550  016701  176346                  MOV·    TD$OTR,R1       ;READ·OUTPUT·
000554  004767  000000G·                 JSR·    PC,UNPK·        ;CONVERT
000560  004767  000000G·                 JSR·    PC,CONSOL·      ;PRINT·IT·
000564  000750                          BR      8$              ;TRY·AGAIN·
                                ;
000566  005726                  RUST:   TST·    (SP)+           ;RESTORE SP·
000570                          RU1X:
000570  042767  000000C·000000G·         BIC·    #<QP+QN>,BASE·  ;CLEAR·RUN·FLAGS·
000576  004767· 000000G·                 JSR·    PC,KILL·        ;KILL·AST·(IF·THERE·IS·ONE)
000602· 000167  000000G·                 JMP·    COMXX·
                                ;
000001                                  .END·
```

```
ALUCKE= 040000
ALUOE = 004000
A01   = 010000
BASE  = ****** GX
BITVAL= 000000
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BTOVER= ****** GX
BTRANS= ****** GX
BYTE0 = 000000
BYTE1 = 000001
BYTE10= 000012
BYTE11= 000013
BYTE12= 000014
BYTE13= 000015
BYTE14= 000016
BYTE15= 000017
BYTE16= 000020
BYTE17= 000021
BYTE18= 000022
BYTE19= 000023
BYTE2 = 000002
BYTE20= 000024
BYTE21= 000025
BYTE22= 000026
BYTE23= 000027
BYTE24= 000030
BYTE25= 000031
BYTE26= 000032
BYTE27= 000033
BYTE28= 000034
BYTE29= 000035
BYTE3 = 000003
BYTE30= 000036
BYTE31= 000037
BYTE32= 000040
BYTE33= 000041
BYTE34= 000042
BYTE35= 000043
BYTE36= 000044
BYTE37= 000045
BYTE38= 000046
BYTE39= 000047
BYTE4 = 000004
BYTE40= 000050
BYTE41= 000051
BYTE42= 000052
BYTE43= 000053
BYTE44= 000054
BYTE45= 000055
BYTE46= 000056
BYTE47= 000057
BYTE48= 000060
BYTE49= 000061
BYTE5 = 000005
BYTE50= 000062
BYTE51= 000063
BYTE52= 000064
BYTE53= 000065
BYTE54= 000066
BYTE55= 000067
BYTE56= 000070
BYTE57= 000071
BYTE58= 000072
BYTE59= 000073
BYTE6 = 000006
BYTE60= 000074
BYTE61= 000075
BYTE62= 000076
BYTE63= 000077
BYTE64= 000100
BYTE65= 000101
BYTE66= 000102
BYTE67= 000103
BYTE68= 000104
BYTE69= 000105
BYTE7 = 000007
BYTE70= 000106
BYTE71= 000107
BYTE72= 000110
BYTE73= 000111
BYTE74= 000112
BYTE75= 000113
BYTE76= 000114
BYTE77= 000115
BYTE78= 000116
BYTE79= 000117
BYTE8 = 000010
BYTE80= 000120
BYTE81= 000121
BYTE82= 000122
BYTE83= 000123
BYTE84= 000124
BYTE85= 000125
BYTE86= 000126
BYTE87= 000127
BYTE88= 000130
BYTE89= 000131
BYTE9 = 000011
BYTE90= 000132
BYTE91= 000133
BYTE92= 000134
BYTE93= 000135
BYTE94= 000136
BYTE95= 000137
BYTE96= 000140
BYTE97= 000141
BYTE98= 000142
BYTE99= 000143
BYTVAL= 000144
CBKALL= 001000
CBKCLK= 000400
CNOBRE= 100000
COMXX = ****** GX
CONSOL= ****** GX
CPCCEN= 010000
CPREAD= 040000
CPWRTE= 020000
CSADRD= 000004
CSEOCI= 100000
CSOE  = 000040
CSWRTE= 000100
DATALN= ****** GX
DBR.RD= 000001
DB$CPP= 001457
DB$SPT= 000026
DB$TPC= 000023
DISPGS= 100000
DMAAWR= 000005
DMARRD= 000003
DMARWR= 000004
DTBL  = ****** GX
ENBR  = 010000
ERR18 = ****** GX
ERR3  = ****** GX
FIND  = ****** GX
HANG  = ****** GX
IHIGH = ****** GX
ILOW  = ****** GX
KILL  = ****** GX
LOC.EN= 000100
LOC.WA= 040000
LOC.WB= 100000
LOOP  = ****** GX
LOOPR = ****** GX
MAREN1= 000001
MAREN2= 004000
MARLOD= 010000
MAROUT= 000002
MAR.LO= 002000
MAR.OU= 000040
MBKALL= 001000
MBKCLK= 000400
MMADRD= 000100
MMLEFT= 000002
MMOE  = 000004
MMWRTE= 000010
MNOBRE= 100000
MREN1 = 000001
MREN2 = 020000
MSTR2 = ****** GX
MSYN  = 000040
N     = 000144
PLB   = 000010
PLC   = 000020
PLD   = 000030
PLRWR = 000200
PLR.EN= 000200
PRINT = ****** GX
QN    = ****** GX
QP    = ****** GX
QR$CR1= 176420
QR$CR2= 176422
QR$LBR= 176424
Q$ATTN= 000100
Q$BCL = 000001
Q$CCCP= 000040
Q$CHB = 000400
Q$CHRL= 000200
Q$CLR = 000040
Q$CNC = 030000
Q$CP  = 000060
Q$CPCC= 000010
Q$CP2 = 000260
Q$CSC = 010000
Q$CSEL= 000360
Q$CSET= 000002
Q$CSP = 020000
Q$DMA = 000001
Q$ENBK= 040000
Q$ENDP= 020000
Q$FAL = 004000
Q$FC  = 000045
Q$FO  = 000044
Q$FP  = 000046
Q$HBF = 000002
Q$ICP = 000006
Q$IHB = 000003
Q$IHRL= 000002
Q$IMRP= 000007
Q$LBD = 001000
Q$LBDP= 001001
Q$LBP = 000001
Q$LDCD= 000003
Q$LDMD= 000004
Q$LDPP= 002000
Q$LHP = 010000
Q$MNC = 140000
Q$MR  = 000052
Q$MRP = 000040
Q$MRP2= 000240
Q$MSC = 040000
Q$MSET= 000004
Q$MSP = 100000
Q$NCLK= 176000
Q$PP  = 000100
Q$PPSW= 000320
Q$PP2 = 000300
Q$QHLT= 000013
Q$QL  = 000043
Q$QLA = 000053
Q$QLB = 000054
Q$QLR = 000001
Q$QW  = 000042
Q$RDCD= 000005
Q$RDMD= 000006
Q$REBK= 001000
Q$RNC = 006000
Q$RSC = 004000
Q$RSET= 000010
Q$SM  = 100000
Q$SP  = 000120
Q$SP2 = 000340
RGQ.EN= 000200
RGQ.VA= 020000
RIDE  = ****** GX
RUNC    000144RG  002
RUQN    000134RG  002
RUQP    000124RG  002
RUSC    000070RG  002
RUST    000566R   002
RUi     000020RG  002
RUiLN = 000004 G
RUiTBL  000000RG  002
RUiX    000570R   002
SCAN  = ****** GX
SEQ.CI= 000010
S$CLR = 000000
S$LA  = 000001
S$QB  = 000005
S$QR  = 000006
S$QX  = 000004
S$SR  = 000007
S$S1  = 000010
S$S2  = 000014
TD$CTR= 176370
TD$CTW= 176360
TD$INL= 004000
TD$MEM= 000270
TD$OAR= 176344
TD$OTR= 176346
TD$QRD= 000274
TD$SW = 176376
TD$TAR= 176372
TD$TAW= 176362
TD$TDR= 176374
TD$TDW= 176364
TLOW  = ****** GX
T$AD  = 000020
T$BA  = 000002
T$BD  = 000010
```

```
TRUN.   MACRO M1110  27-MAR-80 15:40  PAGE 5-5
SYMBOL TABLE

T$BSO = 100000          T$SCLK = 020000         WORD26 = 000064         WORD51 = 000146         WORD77 = 000232
T$BT  = 000020          T$SEG1 = 000000         WORD27 = 000066         WORD52 = 000150         WORD78 = 000234
T$BTAR = 000030         T$SEG2 = 000001         WORD28 = 000070         WORD53 = 000152         WORD79 = 000236
T$BTD = 002000          T$SEG3 = 000002         WORD29 = 000072         WORD54 = 000154         WORD8 = 000020
T$CD  = 000100          T$SO  = 000001          WORD3 = 000006          WORD55 = 000156         WORD80 = 000240
T$CLK = 002000          T$UBUS = 100000         WORD30 = 000074         WORD56 = 000160         WORD81 = 000242
T$DISK = 000200         T$1CLK = 000400         WORD31 = 000076         WORD57 = 000162         WORD82 = 000244
T$DRD = 000004          T$8BEN = 000020         WORD32 = 000100         WORD58 = 000164         WORD83 = 000246
T$EMEM = 010000         UBD.IN = 000020         WORD33 = 000102         WORD59 = 000166         WORD84 = 000250
T$FSAA = 000000         UNPK  = ****** GX       WORD34 = 000104         WORD6 = 000014          WORD85 = 000252
T$FSAB = 000004         WORD0 = 000000          WORD35 = 000106         WORD60 = 000170         WORD86 = 000254
T$FSAC = 000014         WORD1 = 000002          WORD36 = 000110         WORD61 = 000172         WORD87 = 000256
T$FSB2 = 000010         WORD10 = 000024         WORD37 = 000112         WORD62 = 000174         WORD88 = 000260
T$IB  = 000026          WORD11 = 000026         WORD38 = 000114         WORD63 = 000176         WORD89 = 000262
T$IBAR = 000024         WORD12 = 000030         WORD39 = 000116         WORD64 = 000200         WORD9 = 000022
T$IBE = 020000          WORD13 = 000032         WORD4 = 000010          WORD65 = 000202         WORD90 = 000264
T$IBF = 040000          WORD14 = 000034         WORD40 = 000120         WORD66 = 000204         WORD91 = 000266
T$ICD = 000040          WORD15 = 000036         WORD41 = 000122         WORD67 = 000206         WORD92 = 000270
T$MODE = 004000         WORD16 = 000040         WORD42 = 000124         WORD68 = 000210         WORD93 = 000272
T$OB  = 000036          WORD17 = 000042         WORD43 = 000126         WORD69 = 000212         WORD94 = 000274
T$OBE = 004000          WORD18 = 000044         WORD44 = 000130         WORD7 = 000016          WORD95 = 000276
T$OBF = 010000          WORD19 = 000046         WORD45 = 000132         WORD70 = 000214         WORD96 = 000300
T$OBRA = 000034         WORD2 = 000004          WORD46 = 000134         WORD71 = 000216         WORD97 = 000302
T$OBWA = 000032         WORD20 = 000050         WORD47 = 000136         WORD72 = 000220         WORD98 = 000304
T$OUTA = 100000         WORD21 = 000052         WORD48 = 000140         WORD73 = 000222         WORD99 = 000306
T$RBDO = 000200         WORD22 = 000054         WORD49 = 000142         WORD74 = 000224         WRDVAL = 000310
T$RNB = 000040          WORD23 = 000056         WORD5 = 000012          WORD75 = 000226         XTREAD = 001000
T$RSET = 040000         WORD24 = 000060         WORD50 = 000144         WORD76 = 000230         XTWRTE = 000400
T$SC  = 000022          WORD25 = 000062

. ABS.  000000     000
        000000     001
TRUN    000606     002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3221 WORDS  ( 13 PAGES)
DYNAMIC MEMORY:  3860 WORDS  ( 14 PAGES)
ELAPSED TIME:  00:00:45
TRUN,TRUN/-SP=[20,1]IM,[20,1]TRUN
```

```
TASK.   NAME.   : ...AID.
PARTITION NAME. : GEN
IDENTIFICATION. : 08
TASK. UIC.      : [20,3].
STACK.   LIMITS: 040216 041215 001000 00512.
PRG XFR ADDRESS: 051734
TASK ATTRIBUTES: AL,CP.
TOTAL ADDRESS WINDOWS: 2.
TASK. IMAGE. SIZE.  : 8032. WORDS.
TASK ADDRESS LIMITS: 040000 077253
R-W DISK BLK LIMITS: 000042 000110 000047 00039.


AIDTD.TSK;3 OVERLAY DESCRIPTION:

BASE.   TOP.        LENGTH.
----    ---         ------
040000  071007  031010  12808.       TMAIN.
071010  072077  001070  00568.            TREG.
071010  073353  002344  01252.            TMEM.
071010  077253  006244  03236.            TDATA.
071010  071617  000610  00392.            TRUN.
```

*** ROOT SEGMENT: TMAIN

R/W MEM LIMITS: 040000 071007 031010 12808.
DISK BLK LIMITS: 000042 000073 000032 00026.

MEMORY ALLOCATION SYNOPSIS:

SECTION			TITLE	IDENT	FILE
. BLK.:	(RW,I,LCL,REL,CON)	041216 021300 08896.			
		041216 014224 06292.	TMAIN		TMAIN.OBJ;1
$$ALER:	(RW,I,LCL,REL,CON)	062516 000024 00020.			
$$ALVC:	(RW,D,LCL,REL,CON)	062542 000130 00088.			
$$AUTD:	(RW,I,LCL,REL,OVR)	062672 000130 00088.			
$$FSR1:	(RW,D,GBL,REL,OVR)	063022 002040 01056.			
		063022 002040 01056.	TMAIN		TMAIN.OBJ;1
$$FSR2:	(RW,D,GBL,REL,CON)	065062 000104 00068.			
$$MRKS:	(RO,I,LCL,REL,OVR)	070540 000076 00062.			
$$OVDT:	(RW,D,LCL,REL,OVR)	065166 000020 00016.			
$$OVRS:	(RW,I,LCL,ABS,CON)	000000 000000 00000.			
$$RDSG:	(RO,I,LCL,REL,OVR)	070636 000150 00104.			
$$RESL:	(RW,I,LCL,REL,CON)	065206 003232 01690.			
$$RESM:	(RW,I,LCL,REL,CON)	132000 007656 04014.			
$$RGDS:	(RW,D,LCL,REL,CON)	070440 000000 00000.			
$$RTS.:	(RW,I,GBL,REL,OVR)	070440 000002 00002.			
$$SGD0:	(RW,D,LCL,REL,OVR)	070442 000000 00000.			
$$SGD1:	(RW,D,LCL,REL,CON)	070442 000074 00060.			
$$SGD2:	(RW,D,LCL,REL,OVR)	070536 000002 00002.			
$$WNDS:	(RW,D,LCL,REL,CON)	070540 000000 00000.			

GLOBAL SYMBOLS:

AHIGH	041424-R	CLOW	041436-R	EFN.1	000001	ERR21	055026-R	INCVAL	041376-R	OLOW	041446-R	SCAN	054224-R		
ALOW	041426-R	CMILUN	000002	ENDMEM	055002-R	ERR22	055022-R	IO.DET	002000	ONCE	000010	SELTST	055156-R		
APLACE	041234-R	CNUM	000015	ENDTST	054772-R	ERR23	055016-R	IO.RVB	010400	OUT	002000	STAT	041222-R		
ASTFLG	000020	COMXX	051740-R	ENFILE	054776-R	ERR3	055136-R	IO.WVB	011000	OUT1	055012-R	STOP	055006-R		
BASE	041232-R	CONSOL	054610-R	EN1	062632-R	ERR4	055132-R	KILL	054124-R	PACK	054434-R	ST1	062552-R		
BASEL	055162-R	COUNT	041404-R	ERR1	055146-R	ERR5	055126-R	LDCNTL	052440-R	PDATA	052620-R	THIGH	041450-R		
BHIGH	041430-R	CTBL	041454-R	ERR10	055102-R	ERR6	055122-R	LD1	062562-R	PRCNTL	053032-R	TLOW	041452-R		
BINWD	041230-R	DATALN	041406-R	ERR11	055076-R	ERR7	055116-R	LI1	062642-R	PRDATA	053126-R	TRFDB	051556-R		
BLOW	041432-R	DATA1	041416-R	ERR12	055072-R	ERR8	055112-R	LOOP	000004	PRINT	047072-R	TRLUN	000004		
BTEND	046666-R	DATA2	041420-R	ERR13	055066-R	ERR9	055106-R	LOOPR	054004-R	PR1	062572-R	TR1	062652-R		
BTOVER	046670-R	DATA3	041422-R	ERR14	055062-R	FIND	054316-R	LPTST	055152-R	QN	001000	TR6TBL	046266-R		
BTRANS	046466-R	DA1	062612-R	ERR15	055056-R	GCMBUF	041236-R	LUN.TT	000001	QP	000400	TXTFDB	051340-R		
BUFFSA	052044-R	DBLD	053312-R	ERR16	055052-R	GCMLEN	041360-R	MEND	041374-R	RE1	062542-R	TXTLUN	000003		
BUFSET	052232-R	DF1	062622-R	ERR17	055046-R	GCMPNT	041362-R	MSTRT	041370-R	RIDE	000100	UNPK	054544-R		
BUFS2	052222-R	DPLUS	041410-R	ERR18	055042-R	GCONLY	053726-R	MSTR2	041372-R	RP	000200	UNLIM	041462-R		
CHIGH	041434-R	DSAVE	043540-R	ERR19	055036-R	HANG	054040-R	OFFSET	053224-R	RSPCNT	041400-R	WDWORDS	041416-R		
CHLEN	041414-R	DSEND	046264-R	ERR2	055142-R	IHIGH	041440-R	OFF6	041412-R	RTNPT	041366-R	$DIV	007146		
CH1	062602-R	DTBL	041540-R	ERR20	055032-R	ILOW	041442-R	OHIGH	041444-R	RU1	062662-R	$MUL	007116		

*** SEGMENT: TREG

R/W MEM LIMITS: 071010 072077 001070 00568.
DISK BLK LIMITS: 000074 000075 000002 00002.

MEMORY ALLOCATION SYNOPSIS:

SECTION		TITLE	IDENT	FILE
. BLK:	(RW,I,LCL,REL,CON) 071010 000000 00000.			
TREG	(RW,I,LCL,REL,CON) 071010 001070 00568.			
	071010 001070 00568.	TREG		TREG.OBJ;1
$$ALVC:	(RW,D,LCL,REL,CON) 072100 000000 00000.			
$$RTS	(RW,I,GBL,REL,OVR) 070440 000002 00002.			

GLOBAL SYMBOLS:

REAA	071536-R	REIA	071602-R	RESC	071670-R	RE1LN	000013	STBT	071356-R	STO	071332-R	ST1TBL	071010-R
REBA	071552-R	REMS	071726-R	RESW	071720-R	RE1TBL	071050-R	STC	071254-R	STSC	071402-R		
RECA	071566-R	REOA	071624-R	RETA	071646-R	STA	071234-R	STCR	071426-R	ST1	071124-R		
RECR	071712-R	REQR	071750-R	RE1	071456-R	STB	071244-R	STI	071306-R	ST1LN	000010		

*** SEGMENT: TMEM

/W MEM LIMITS: 071010 073353 002344 01252.
?ISK BLK LIMITS: 000076 000100 000003 00003.

MEMORY ALLOCATION SYNOPSIS:

SECTION				TITLE	IDENT	FILE
. BLK:	(RW,I,LCL,REL,CON)	071010	000000 00000.			
TMEM	:(RW,I,LCL,REL,CON)	071010	002344 01252.			
		071010	002344 01252.	TMEM		TMEM.OBJ;1
$$ALVC:	(RW,D,LCL,REL,CON)	073354	000000 00000.			
$$RTS	:(RW,I,GBL,REL,OVR)	070440	000002 00002.			

GLOBAL SYMBOLS:

LDAM	071342-R	LDCM	071432-R	LD1	071070-R	PRAM	072660-R	PRCM	072750-R	PR1	072372-R
LDBM	071376-R	LDIB	071466-R	LD1LN	000006	PRBM	072714-R	PRIB	073004-R	PR1LN	000006
LDBT	072122-R	LDOB	071664-R	LD1TBL	071010-R	PRBT	073220-R	PROB	073102-R	PR1TBL	071040-R

*** SEGMENT: TDATA

R/W MEM LIMITS: 071010 077253 006244 03236.
DISK BLK LIMITS: 000101 000107 000007 00007.

MEMORY ALLOCATION SYNOPSIS:

SECTION				TITLE	IDENT	FILE
. BLK.	(RW,I,LCL,REL,CON)	071010	000000 00000.			
TDATA	(RW,I,LCL,REL,CON)	071010	002364 01268.			
		071010	002364 01268.	TDATA		TDATA.OBJ;1
$$ALVC	(RW,D,LCL,REL,CON)	073374	000000 00000.			
$$RESL	(RW,I,LCL,REL,CON)	073374	003660 01968.			
$$RTS	(RW,I,GBL,REL,OVR)	070440	000002 00002.			

GLOBAL SYMBOLS:

CHBT	071406-R	CHITBL	071010-R	DFI	072326-R	LIBT	072102-R	LIITBL	071020-R	TRILN	000002
CHIB	071270-R	DAI	071050-R	DFILN	000002	LIIB	071702-R	TRBT	073236-R	TRITBL	071040-R
CHI	071160-R	DFBT	072606-R	DFITBL	071030-R	LII	071562-R	TRIB	073144-R		
CHILN	000002	DFIB	072376-R	ENI	072234-R	LIILN	000002	TR1	073064-R		

```
*** SEGMENT: TRUN

R/W MEM  LIMITS: 071010 071617 000610 00392.
DISK BLK LIMITS: 000110 000110 000001 00001.

MEMORY ALLOCATION SYNOPSIS:

SECTION                                          TITLE   IDENT   FILE
-------                                          -----   -----   ----
. BLK. :(RW,I,LCL,REL,CON) 071010 000000 00000.
TRUN   :(RW,I,LCL,REL,CON) 071010 000606 00390.
                           071010 000606 00390.  TRUN            TRUN.OBJ;1
$$ALVC:(RW,D,LCL,REL,CON) 071616 000000 00000.
$$RTS  :(RW,I,GBL,REL,OVR) 070440 000002 00002.


GLOBAL SYMBOLS:

RUNC   071154-R  RUQN   071144-R  RUQP   071134-R  RUSC   071100-R  RU1    071030-R  RU1LN  000004    RU1TBL 071010-R


*** TASK BUILDER STATISTICS:

    TOTAL WORK FILE REFERENCES: 40221.
    WORK  FILE  READS: 0.
    WORK  FILE WRITES: 0.
    SIZE OF CORE POOL: 9802. WORDS (38. PAGES)
    SIZE OF WORK FILE: 8192. WORDS (32. PAGES)

    ELAPSED TIME:00:00:28
```

```
                        .TITLE  QMAIN
                ;
                ;
                ;       HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
                ;       MAIN MODULE
                ;
                ;
                ;       THIS MODULE PASSES CONTROL TO ITS SUB-MODULES (HOR PROCESSOR
                ;       MODULES) BASED ON INFORMATION IN THE COMMAND LINE (TERMINAL
                ;       INPUT).  THE SUB-MODULES CONTINUE TO PARSE THE COMMAND LINE,
                ;       TRANSFERRING CONTROL TO THEIR SUB-MODULES.  THESE SUB-MODULES
                ;       CONTINUE TO PARSE THE COMMAND LINE, TRANSFERRING CONTROL TO
                ;       THEIR SUB-ROUTINES.  THE SUB-ROUTINES DO THE ACTUAL INTERFACING
                ;       WITH THE HARDWARE.  THUS THERE ARE FOUR LEVELS OF CONTROL.  FOR
                ;       EXAMPLE, TAKE THE COMMAND:
                ;       >MR LD MD 0
                ;       THE 'MR' REPRESENTS THE FIRST LEVEL OF CONTROL AND IS PARSED
                ;       BY THE MODULE QMAIN.  THE 'LD' REPRESENTS THE SECOND LEVEL OF
                ;       CONTROL AND IS PARSED BY THE QMAIN SUB-MODULE MRP.  THE 'MD'
                ;       REPRESENTS THE THIRD LEVEL OF CONTROL AND IS PARSED BY THE
                ;       MRP SUB-MODULE MRLD.  THE FOURTH LEVEL OF CONTROL, A SUB-ROUTINE
                ;       OF MRLD, ACTUALLY CONTROLS THE LOADING OF THE HARDWARE.
                ;
                ;
                ;       SUB-MODULES OF QMAIN:
                ;       MRP     MATCH REPORT PROCESSOR
                ;       CP      CONTROL PROCESSOR
                ;       BCE     BUS CONTROL ELEMENT
                ;       PPS     PIPELINED PROCESSORS
                ;       SP      SUBDOCUMENT PROCESSOR
                ;
                ;       SUB-MODULES OF MRP:
                ;       MRLD    LOAD
                ;       MRPR    PRINT
                ;       MRREST  ALL OTHER COMMANDS (EXCEPT MICROCODE DEBUGGING COMMANDS)
                ;       MRBUG   DEBUGGING COMMANDS
                ;
                ;       SUB-MODULES OF CP:
                ;       CPLD    LOAD
                ;       CPPR    PRINT
                ;       CPREST  ALL OTHER COMMANDS (EXCEPT MICROCODE DEBUGGING COMMANDS)
                ;       CPBUG1  SOME MICROCODE DEBUGGING COMMANDS
                ;       CPBUG2  THE REST OF THE DEBUGGING COMMANDS
                ;
                ;       SUB-MODULES OF BCE:
                ;       BCREST  ALL BCE COMMANDS
                ;
                ;       SUB-MODULES OF PPS:
                ;       PPLD    LOAD
                ;       PPPR    PRINT
                ;       PPREST  ALL OTHER COMMANDS
                ;
                ;       SUB-MODULES OF SP:
                ;       SPLD    LOAD
                ;       SPPR    PRINT
                ;       SPREST  ALL OTHER COMMANDS
```

```
                ;       CONTROL IS RETURNED TO QMAIN WHEN ONE OF ITS SUB-MODULES
                ;       ENCOUNTERS AT THE SECOND LEVEL OF CONTROL (SEE ABOVE) A
                ;       STRING THAT IT CANNOT PARSE. THE ASSUMPTION IS THAT THIS
                ;       STRING IS THE MNEMONIC FOR ANOTHER PROCESSOR, REQUIRING A
                ;       TRANSFER OF CONTROL. FOR EXAMPLE, THE COMMAND
                ;       MR>CP LD CS 0
                ;       CANNOT BE PARSED BY MRP ('CP' IS NOT A VALID MRP COMMAND),
                ;       CONTROL IS PASSED TO QMAIN WHICH CAN PARSE 'CP', CONTROL
                ;       WILL THEN BE TRANSFERRED FROM QMAIN TO THE SUB-MODULE CP.
                ;
                ;
                ;       QMAIN ALSO CONTAINS DATA AND SUBROUTINES COMMON TO ALL OF ITS
                ;       SUB-MODULES. THERE ARE ALSO MODULES CONTAINING SUBROUTINES
                ;       FOR THE QMAIN SUB-MODULES. THE SUBROUTINE MODULES ARE MRPSUB,
                ;       CPSUB, AND PPSUB. ROUTINES IN THESE MODULES ARE GLOBAL, ALLOW-
                ;       ING CROSS-USAGE. THUS THE QEX WINDOW MEMORY LOADING ROUTINE
                ;       (LD4QW IN SUB-MODULE PPLD OF PPS) CALLS THE SUBROUTINE
                ;       SEQCS IN MODULE CPSUB AND SUBROUTINE SEQMM IN MODULE MRPSUB.
                ;
                ;
                ;       ASSEMBLY:
                ;       MCR>MAC QMAIN,LP=IM04,QMAIN          FROM [5,3]
                ;
                ;       TASK BUILD:
                ;       1. HQR STAND-ALONE PACK COMMAND FILE AIDQ2.CMD
                ;       AIDQR/AL/CP/DA,AIDQR=QMAIN,MRP,MRLD,MRPR,MRREST,MRBUG,
                ;       CP,CPLD,CPPR,CPREST,CPBUG1,CPBUG2,
                ;       BCE,BCREST,
                ;       PPS,PPLD,PPPR,PPREST,
                ;       MRPSUB,CPSUB,PPSUB,[1,50]RSX11M.STB/SS
                ;       /
                ;       PAR=PAR14K
                ;       ASG=TT0:1:2
                ;       /
                ;       //
                ;
                ;       2. NPIC SYSTEM COMMAND FILE AIDQR.CMD (USING OVERLAYS)
                ;          ********** NOTE **********
                ;          TASK BUILD MUST BE DONE ON NPIC /04
                ;          MAPPING DOES NOT COME OUT RIGHT WHEN TKB IS DONE ON /45
                ;          EG. ADDRESS OF SYSTEM ROUTINE $DIV COMES OUT INCORRECTLY
                ;          ********** NOTE **********
                ;
                ;       AIDQR/AL/CP,AIDQR=AIDQR/MP
                ;       TASK=AIDQR
                ;       PAR=GEN:40000:40000
                ;       ASG=TT0:1:2
                ;       //
                ;
                ;       OVERLAY DESCRIPTION AIDQR.ODL:
                ;               .ROOT   QMAIN-[1,50]RSX11M.STB/SS-*(A,B,C,D)
                ;       A:      .FCTR   MRP-MRPSUB-CPSUB-*(MRLD,MRPR,MRREST,MRBUG)
                ;       B:      .FCTR   CP-MRPSUB-CPSUB-PPSUB-(CPBUG1,CPBUG2,CPREST,CPLD,CPPR)
                ;       C:      .FCTR   BCE-BCREST
                ;       D:      .FCTR   PPS-MRPSUB-CPSUB-PPSUB-(PPLD,PPPR,PPREST)
                ;       .END
```

```
                                        ;
                                        ;
                                        ;       LOCAL DATA AREAS
                                        ;
                                        ;
                                                .MCALL  QIOW$S,QIO$S,EXIT$S,ABRT$S,GCML$,GCMLB$,FSRSZ$,CLEF$S
                                                .MCALL  ASTX$S,RDAF$S,WTSE$S,SETF$S,RQST$S
                                        ;
                                                .GLOBL  IO.WVB,IO.RVB,IO.ATA,IO.DET
                                        ;
                        000160          G.DPRM  ==      000160
                        000001          LUN.TT  ==      1               ;READ/WRITE TT0
                        000001          EFN.1   ==      1               ;EVENT FLAG FOR TT0
                        000002          EFN.2   ==      2
                        000003          EFN.3   ==      3               ;EVENT FLAG FOR HQR INTERRUPTS
                        000004          EFN.4   ==      4               ;EVENT FLAG FOR UNSOLICITED TERMINAL INTERRUPTS
                        000041          EFN.33  ==      33.             ;EVENT FLAG FOR COMMUNICATION WITH HQR LOADER
                        000002          CMILUN  =       2
                                        ;
                        000004          LOOP    ==      4               ;LOOP ON COMMAND
                        000100          ONCE    ==      100             ;PRINT ONE WORD ONLY
                        000200          ASTFLG  ==      200             ;QIO + AST ISSUED
                        001000          RP      ==      1000            ;REPEAT PROMPT
                        002000          OUT     ==      2000            ;CONTROL PRINTING OF MEMORY CONTENTS
                        004000          BREAK   ==      4000            ;BREAKPOINT SET
                        010000          NEQLB   ==      10000           ;NO QLB ERASE
                                        ;
                                        ;
                                                .NLIST  BEX
000000  003654  066420          MYSELF:: .RAD50 /AIDQR/
000004  000000                  TSKTCB:: .WORD  0               ;TCB OF MY TASK
000006  000000                  OLDVEC:: .WORD  0               ;OLD VECTOR AT 274
000010  046531  014732          LOADER: .RAD50  /LOADER/
000014                          EFBUF:: .BLKW   4               ;EVENT FLAG BUFFER
000024                          STAT::  .BLKW   2
000030  000000                  ERWORD:: .WORD  0               ;INDEX VALUE FOR ERROR MESSAGE TABLE
000032  000000                  BINWD:: .WORD   0               ;TARGET FOR NUMERIC CONVERSIONS FROM ASCII
000034  000000                  BASE::  .WORD   0               ;ALL PURPOSE FLAG
000036  000000                  APLACE:: .WORD  0               ;PRELIM BIT SETTINGS FOR CSR 1
000040                          GCMBUF:: .BLKW  41.             ;COMMAND LINE BUFFER
000162  000000                  GCMLEN:: .WORD  0               ;COMMAND LINE LENGTH
000164  000000                  GCMPNT:: .WORD  0               ;COMMAND LINE POINTER
000166  000000                  ASTWRD:: .WORD  0               ;RECEIVER FOR AST CHAR
000170  000000                  RTNPT:: .WORD   0               ;RTN ADDR SAVE AREA
000172  000000                  MSTRT:: .WORD   0               ;START ADDR FOR MEMORY LOADING/PRINTING
000174  000000                  MSTR2:: .WORD   0               ;WORKING ADDR FOR LOAD/PRINT
000176  000000                  MEND::  .WORD   0               ;END ADDR FOR MEMORY LOADING/PRINTING
000200  000000                  INCVAL:: .WORD  0               ;MEMORY INCREMENT VALUE
000202  000000                  RSPCNT:: .WORD  0               ;(COMMAND LINE) RESPONSE COUNT
000204  000000                  UPLIM:: .WORD   0               ;MEMORY UPPER LIMIT
                                        ;
000206                          WWORDS:                         ;LOAD/PRINT VALUES
000206  000000                  DATA1:: .WORD   0
000210  000000                  DATA2:: .WORD   0
000212  000000                  DATA3:: .WORD   0
000214  000000                  DATA4:: .WORD   0
                                        ;
```

```
                                ;       MEMORY LIMITS TABLE
                                ;
000216  000377                 MMHIGH:: .WORD  255.                    ;MRP MICROPGM MEMORY
000220  000000                 MMLOW:: .WORD   0
000222  007777                 MDHIGH:: .WORD  4095.                   ;MRP DATA MEMORY
000224  000000                 MDLOW:: .WORD   0
000226  001777                 CSHIGH:: .WORD  1023.                   ;CP CONTROL STORE
000230  000000                 CSLOW:: .WORD   0
000232  007777                 CDHIGH:: .WORD  4095.                   ;CP DATA MEMORY
000234  000000                 CDLOW:: .WORD   0
000236  077777                 QXHIGH:: .WORD  077777                  ;=X'7FFF' QEX MEMORIES
000240  076000                 QXLOW:: .WORD   076000                  ;=X'7C00'
000242  007777                 FAHIGH:: .WORD  4095.                   ;FAL MEMORIES
000244  000000                 FALOW:: .WORD   0
000246  002000                 LHHIGH:: .WORD  1024.                   ;LHP MEMORIES (ALLOW X'400' = ILLEGAL ADDRESS)
000250  000000                 LHLOW:: .WORD   0
000252  000177                 HLHIGH:: .WORD  127.                    ;HRL BUFFER IN CC MEMORY
000254  000000                 HLLOW:: .WORD   0
000256  000014                 BLHIGH:: .WORD  12.                     ;BCL IN CC MEMORY
000260  000000                 BLLOW:: .WORD   0
000262  000377                 SQHIGH:: .WORD  255.                    ;SP REFERENCE, QLB MEMORIES
000264  000000                 SQLOW:: .WORD   0
000266  000377                 SRHIGH:: .WORD  255.                    ;SP SIDREAD MEMORY
000270  000000                 SRLOW:: .WORD   0
000272  007777                 SDHIGH:: .WORD  4095.                   ;SP SIDMEM MEMORY
000274  000000                 SDLOW:: .WORD   0
                                ;
                                ;       TABLE USED IN CONVERSION FROM ASCII HEX TO BINARY
                                ;
000276                          TRTBL::
        000356'                 .       =.      .+60
000356      000     001     002         .BYTE   0,1,2,3,4,5,6,7,8.,9.
        000377'                 .       =.      TRTBL+101
000377      012     013     014         .BYTE   10.,11.,12.,13.,14.,15.
        000475'                 .       =.      TRTBL+177
                                ;
                                ;       TBALE USED IN CONVERSION FROM BINARY TO ASCII HEX
                                ;
000475      060     061     062 TRTBL2:: .ASCII /0123456789ABCDEF/
                                .EVEN
```

```
                                        ;
                                        ;
                                        ;       FIRST·LEVEL·CONTROL·TABLE·
                                        ;       PROCESSOR·MNEMONICS·AND·ASSOCIATED·QMAIN·SUB-MODULE·
                                        ;       ADDRESSES·
                                        ;
                                        ;
000516                                  FTBL:
000516      115     122                         .ASCII·  /MR/                   ;MATCH·REPORT·PROCESSOR·
000520  000000G·                                .WORD·   MRP·
000522·     103     120                         .ASCII·  /CP/                   ;CONTROL·PROCESSOR·
000524  000000G·                                .WORD·   CP·
000526      102·    103·                        .ASCII·  /BC/                   ;BUS·CONTROL·ELEMENT
000530  000000G·                                .WORD·   BCE·
000532·     120     120                         .ASCII·  /PP/                   ;PIPELINE·PROCESSORS
000534  000000G·                                .WORD·   PPS·
000536      123     120                         .ASCII·  /SP/                   ;SUBDOCUMENT·PROCESSOR·
000540  000000G·                                .WORD·   SPS·
000542·     105     130                         .ASCII·  /EX/                   ;EXIT·
000544  002370'                                 .WORD·   EXIT
        000006                          FNUM·   =·       <.-FTBL>/4
                                        ;
                                        ;
                                        ;       PRINT·LINE·
                                        ;
000546      015     012                         .BYTE·   15.12·                 ;PRECEDE·PRINT·LINE·WITH·CRLF·
000550                                  PRINT::
        000116                                  .REPT·   78.
                                                .BYTE·   40
                                                .ENDR·
                                        ;
                                        ;
                                        ;       TABLE·OF·MESSAGES·
                                        ;
                                        ;
000666      000                                 .BYTE·   0
000667      015     012     015                 .BYTE·   15.12,15.12·
000673      105     130     111                 .ASCII·  /EXIT·HOR·DEBUGGING·AIDS/·
000722·     015     012·    000                 .BYTE·   15.12,0
000725      015     012·                        .BYTE·   15.12·
000727      124     111·    115                 .ASCIZ·  /TIME·OUT·ON·FREE·RUN/·
000754      015     012                         .BYTE·   15.12·
000756      105     116·    104                 .ASCIZ·  /END·OF·FILE·REACHED/·
001002·     015     012·                        .BYTE·   15.12·
001004      105     116     104                 .ASCIZ·  /END·OF·MEMORY·REACHED/·
001032·     015     012·                        .BYTE·   15.12·
001034      105     116·    124                 .ASCIZ·  /ENTER·ANY·CHARACTER·TO·EXIT·LOOP/
001075      015     012·    015                 .BYTE·   15.12,15.12,15.12·
001103      110     101     122·                .ASCII·  /HARDWARE·QUERY·RESOLVER·MANUAL·DEBUGGING·AIDS/·
001160      015     012·    000                 .BYTE·   15.12,0
001163      015     012                         .BYTE·   15.12·
001165      111     116     126                 .ASCIZ·  /INVALID·COMMAND/·
001205      015     012                         .BYTE·   15.12·
001207      111     116·    126                 .ASCIZ·  /INVALID·LOOP·OPTION/·
001233      015     012                         .BYTE·   15.12·
001235      111     116     126                 .ASCIZ·  /INVALID·UPPER·MEMORY·LIMITS/·
001271      015     012                         .BYTE·   15.12·
```

```
270 001273      111     116     126             .ASCIZ  /INVALID LOWER MEMORY LIMITS/
271 001327      015     012                     .BYTE   15,12
272 001331      111     116     103             .ASCIZ  /INCORRECT COMMAND CHARACTER COUNT/
273 001373      015     012                     .BYTE   15,12
274 001375      111     116     126             .ASCIZ  /INVALID MEMORY MNEMONIC/
275 001425      015     012                     .BYTE   15,12
276 001427      111     116     126             .ASCIZ  /INVALID REGISTER MNEMONIC/
277 001461      015     012                     .BYTE   15,12
278 001463      111     116     126             .ASCIZ  /INVALID NUMERIC VALUE/
279 001511      015     012                     .BYTE   15,12
280 001513      115     111     123             .ASCIZ  /MISSING OPERAND/
281 001533      015     012                     .BYTE   15,12
282 001535      115     111     123             .ASCIZ  /MISSING COMMAND/
283 001555      015     012                     .BYTE   15,12
284 001557      116     117     116             .ASCIZ  /NON-EXISTENT PROCESSOR/
285 001606      015     012                     .BYTE   15,12
286 001610      120     122     117             .ASCIZ  /PROCESSOR MNEMONIC MUST PRECEED NEXT COMMAND/
287 001665      377             ASCIZ:: .BYTE   377
288                             .EVEN
289                             .LIST   BEX
290                             .NLIST  CND
291                     ;
292                     ;
293                     ;       COMMAND LINE MACRO
294                     ;
295                     ;
296 001666              GCMBLK::        GCMLB$  2,,GCMBUF,CMILUN
297 002174                      FSRSZ$  1
```

```
                                        ;
                                        ;
                                        ;       ENTER HERE
                                        ;
                                        ;
002174                                  START:
002174                                          CALL    OUT1                    ;ISSUE INFORMATION MESSAGE
002200  016767  000000G 175576                  MOV     $TKTCB,TSKTCB           ;SAVE MY TCB
002206  013767  000274  175572                  MOV     @#274,OLDVEC            ;SAVE VECTOR AT 274
002214  012737  003330' 000274                  MOV     #BPTISR,@#274           ;MOVE IN MY INTERRUPT HANDLER ADDR
                                        ;
                                        ;       START OFF WITH MASTER RESET, NO-CLOCKS IN CSR #1
                                        ;
002222  012746  177777                          MOV     #177777,-(SP)           ;CLEAR CSR1
002226  012746  000010                          MOV     #Q$RSET,-(SP)           ;SET RESET
002232                                          CALL    CSR1                    ;RESET HOR
002236  012746  000010                          MOV     #Q$RSET,-(SP)           ;CLEAR RESET
002242  012746  176000                          MOV     #<Q$NCLK>,-(SP)         ;SET NO CLKS
002246                                          CALL    CSR1                    ;MOVE TO CSR1
002252  012746  000040                          MOV     #Q$CLR,-(SP)            ;REINHIBIT FAL PROCEESOR
002256                                          CALL    PPCR
002262  012746  000000                          MOV     #S$CLR,-(SP)            ;CLEAR SP ADDRESS SELECT
002266                                          CALL    SPCR
                                        ;
                                        ;       PUT OUT INITIAL PROMPT, LOCATE 2-CHAR PROCESSOR/COMMAND MNEMONIC,
                                        ;       EG. IF THE COMMAND LINE READS:
                                        ;       >MR LD MD 0
                                        ;       LOCATE THE 'MR'
                                        ;
                                        ;       MR      MATCH REPORT PROCESSOR
                                        ;       CP      CONTROL PROCESSOR
                                        ;       BC      BUS CONTROL ELEMENT
                                        ;       PP      PIPELINED PROCESSORS
                                        ;       SP      SUBDOCUMENT PROCESSOR
                                        ;       EX      EXIT DEBUGGING AIDS (COMMAND)
                                        ;
002272                                  RESEL:
002272  012767  020040  177550                  MOV     #20040,GCMBLK+G.DPRM+2  ;CLEAR PROCESSOR NAME
002300                                          CALL    GCONLY                  ;ISSUE GCML
002304                                          CALL    FIND                    ;FIND THE PROCESSOR MNEMONIC
002310  103003                                  BCC     1$                      ;OK, CONTINUE
002312                                          CALL    ERR2                    ;NOTHING IN COMMAND LINE
002316  000765                                  BR      RESEL
002320  022700  000002                  1$:     CMP     #2,R0                   ;COMMANDS ARE 2 CHARS
002324  001403                                  BEQ     COMXX
002326                                          CALL    ERR8                    ;INCORRECT CHAR COUNT
002332  000757                                  BR      RESEL                   ;TRY AGAIN
                                        ;
                                        ;
                                        ;       TOP OF PROCESSOR LOOP
                                        ;       MATCH PROCESSOR NAME AGAINST TABLE OF NAMES + CONTROL ROUTINE
                                        ;       ADDRESSES
                                        ;
                                        ;       R1 -> PROCESSOR MNEMONIC IN THE COMMAND LINE
                                        ;
                                        ;
002334  012700  000006                  COMXX:: MOV     #FNUM,R0                ;R0 = NUMBER OF PROCESSORS
```

```
356 002340  012702· 000516'                MOV·    #FTBL,R2·          ;R2·-> TABLE·OF·PROCESSOR·MNEMONICS·
357 002344                                  CALL·   SCAN               ;FIND·MATCH·IN·TABLE
358 002350  103005                          BCC·    3$                 ;OK, CONTINUE·
359 002352·                                 CALL·   ERR2               ;COMMAND·NOT·IN·TABLE·
360 002356                                  CALL·   ERR1               ;WHAT·TO·DO·NEXT·
361 002362· 000743                          BR      RESEL·             ;TRY·AGAIN
362·                                ;
363                                 ;
364                                 ;       TRANSFER·CONTROL·TO·ONE·OF·THE·QMAIN·SUB-MODULES·
365                                 ;       FOR·FURTHER·COMMAND·LINE·PARSING
366                                 ;
367                                 ;       R1 -> SUB-MODULE·(CONTROL·ROUTINE) ADDRESS
368                                 ;
369                                 ;
370 002364  000171  000000··        3$:     JMP·    @(R1)
371                                 ;
372·                                ;
373                                 ;       EXIT·PROGRAM
374                                 ;
375                                 ;
376 002370                          EXIT:
377 002370  016737  175412  000274          MOV·    OLDVEC,@#274       ;RESTORE·ORIGINAL VECTOR·CONTENTS·
378 002376                                  CALL·   ENDTST·            ;PUT·OUT·END·OF·RUN·MESSAGE·
379 002402·                                 EXIT$S·
```

```
                                        ;
                                        ;
                                        ;       ROUTINE TO PLACE A VALUE INTO CONTROL/STATUS
                                        ;       REGISTER 1 (176420).
                                        ;
                                        ;       READ THE CURRENT VALUE OF CSR1, CLEAR THE BITS
                                        ;       AT 4(SP) IN A CSR1 WORK AREA, SET THE BITS AT 2(SP)
                                        ;       IN THE SAME WORK AREA, RE-WRITE CSR1.
                                        ;
                                        ;       INPUT:
                                        ;       2(SP)    - BITS TO BE SET IN CSR1
                                        ;       4(SP)    - BITS TO BE CLEARED IN CSR1
                                        ;
                                        ;
002410                                  CSR1::
002410  016767  176420  175420                   MOV     QR$CR1,APLACE           ;GET THE CURRENT VALUE
002416  046667  000004  175412                   BIC     4(SP),APLACE            ;CLEAR FIRST
002424  056667  000002  175404                   BIS     2(SP),APLACE            ;THEN SET
002432  016767  175400  176420                   MOV     APLACE,QR$CR1           ;NOW RETURN IT
002440  011666  000004                           MOV     (SP),4(SP)              ;MOVE RETURN ADDR TO TOP OF STACK
002444  022626                                   CMP     (SP)+,(SP)+             ;BUMP STACK POINTER PAST ARGS
002446                                          RETURN                          ;LEAVE
```

```
                                ;
                                ;
                                ;       CALL HARDWARE QUERY RESOLVER LOADER.
                                ;
                                ; ********** NOTE **********
                                ;       WHEN ON THE NPIC SYSTEM DO NOT USE CL IF CCIN AND CCOUT
                                ;       ARE RUNNING.
                                ; ********** NOTE **********
                                ;
                                ;
                                ;       CLEAR GLOBAL EVENT FLAG.
                                ;
002450                          CL::
002450                                  CLEF$S  #EFN.33
                                ;
                                ;       REQUEST LOADER.
                                ;
002462                                  RQST$S  #LOADER
                                ;
                                ;       WAIT FOR 'LOADER' TO SET EVENT FLAG.
                                ;
002524                                  WTSE$S  #EFN.33
002536                                  RETURN
```

```
                                ;
                                ;
                                ;       SET UP BEFORE PROMPTING ON ANY 'LD' COMMAND OR BEFORE
                                ;       EXECUTING ANY 'PR' COMMAND.
                                ;       CALLED BY ROUTINES IN MMLD, MMPR, CPLD, CPPR, PPLD,
                                ;       PPPR. THE FIELDS SET UP HERE ARE USED IN THE QMAIN
                                ;       PROMPTING SUBROUTINE PDATA AND THE PRINT ROUTINE PRDATA.
                                ;
                                ;       INPUT:
                                ;       2(SP)   LOAD/PRINT START ADDRESS
                                ;       4(SP)   LOWER MEMORY LIMIT
                                ;       6(SP)   UPPER MEMORY LIMIT
                                ;
                                ;       OUTPUT:
                                ;       INCVAL  - VALUE BY WHICH MEMORY ADDRESS INCREMENTS
                                ;       MEND    - ADDRESS OF LAST WORD IN MEMORY TO BE LOADED/PRINTED.
                                ;       RSPCNT  - NUMBER OF WORDS TO EXPECT IN COMMAND LINE (MEMORY WIDTH)
                                ;
                                ;       C-BIT CLEAR     NO ERROR (START ADDRESS IS IN RANGE)
                                ;       C-BIT SET       ERROR IN RANGE
                                ;
                                ;
                                ;       MEMORY          INCVAL          RSPCNT
                                ;       MM              1               2
                                ;       MD              1               1
                                ;       CS              1               4
                                ;       CD              1               1
                                ;       HL              2               1
                                ;       BL              2               1
                                ;       QW              1               1
                                ;       QL              1               1
                                ;       FP              1               1
                                ;       FC              1               1
                                ;       QR              1               1
                                ;       Q0              1               1
                                ;       Q1              1               1
                                ;       Q2              1               1
                                ;       QX              1               1
                                ;       QB              1               1
                                ;       SR              1               1
                                ;       S1              1               3
                                ;       S2              1               3
                                ;
002540  012767  000004  175434  BUFS4:: MOV     #4.RSPCNT       ;NUMBER OF WORDS TO PROMPT
002546  012767  000001  175424          MOV     #1.INCVAL       ;BUFFERS INC BY 1
002554  000433                  BUFS3:: BR      SET
002556  012767  000003  175416          MOV     #3.RSPCNT       ;NUMBER OF WORDS TO PROMPT
002564  012767  000001  175406          MOV     #1.INCVAL       ;BUFFERS INC BY 1
002572  000424                  BUFS2:: BR      SET
002574  012767  000002  175400          MOV     #2.RSPCNT       ;PROMPT 2 WORDS
002602  012767  000001  175370          MOV     #1.INCVAL       ;BUFFERS INC BY 1
002610  000415                  BUFSET:: BR     SET
002612  012767  000001  175362          MOV     #1.RSPCNT       ;PROMPT 1 WORD
002620  012767  000001  175352          MOV     #1.INCVAL       ;BUFFERS INC BY 1
002626  000406                  BUFSM:: BR      SET
002630  012767  000001  175344          MOV     #1.RSPCNT       ;PROMPT 1 WORD
```

```
485 002636  012767  000002  175334          MOV     #2,INCVAL       ;BUFFERS INC BY 2
486                                         ;
487 002644  022767  177777  175324  SET:    CMP     #-1,MEND        ;MEMORY END ADDR SET UP
488 002652  001003                          BNE     10$             ;YES
489 002654  016667  000004  175314          MOV     4(SP),MEND      ;SET END ADDR = MEMORY UPPER LIMIT
490 002662  026766  175304  000002  10$:    CMP     MSTRT,2(SP)     ;IS LOAD ADDRESS IN RANGE (LOW)
491 002670  103003                          BHIS    1$              ;OK, CONTINUE
492 002672                                  CALL    ERR9            ;OUT OF RANGE
493 002676  000424                          BR      BUFCX2
494 002700  026766  175272  000004  1$:     CMP     MEND,4(SP)      ;IS ADDR IN RANGE (HIGH)
495 002706  101403                          BLOS    2$              ;YES, CONTINUE
496 002710                                  CALL    ERR10           ;OUT OF RANGE
497 002714  000415                          BR      BUFCX2
498 002716  026767  175250  175252  2$:     CMP     MSTRT,MEND      ;IS START ADDR LOWER THAN END ADDR
499 002724  101403                          BLOS    3$              ;YES
500 002726                                  CALL    ERR10           ;OUT OF RANGE
501 002732  000406                          BR      BUFCX2
502
503 002734  011666  000004          3$:     MOV     (SP),4(SP)      ;MOVE RETURN ADDRESS
504 002740  062706  000004                  ADD     #4,SP           ;ADJUST SP (FOR MEM LIMITS)
505 002744  000241                          CLC
506 002746  000405                          BR      BUFXX2          ;RETURN
507 002750  011666  000004  BUFCX2: MOV     (SP),4(SP)      ;MOVE RETURN ADDRESS
508 002754  062706  000004                  ADD     #4,SP
509 002760  000261                          SEC
510 002762                          BUFXX2: RETURN
```

```
                                ;
                                ;
                                ;       PROMPTING·CONTROL·FOR·LOADING·ALL·MEMORIES AND·BUFFERS.
                                ;       READ·NUMERIC DATA·FROM THE·COMMAND·LINE, CONVERT·AND STORE·
                                ;       INTO·A·COMMON·BUFFER. EG. IF·THE INITIATING·COMMAND·IS:
                                ;       CP>LD·CS·0
                                ;       THIS·ROUTINE WILL·PUT·OUT·A·PROMPT·AND·EXPECT·4 NUMERIC·
                                ;       VALUES·IN·RETURN:
                                ;       >0000 0000 0000 0000
                                ;       THIS·ROUTINE CONVERTS·THESE·ASCII·HEX·VALUES·INTO·BINARY·
                                ;       AND·STORES·THEM·INTO·DATA1, DATA2, DATA3, DATA4
                                ;
                                ;       INPUT: (SET·UP·BY·QMAIN·SUBROUTINE·BUFSET)
                                ;       MSTR2·   - CURRENT·MEMORY·ADDRESS
                                ;       MEND·    - MEMORY·UPPER ADDRESS·LIMIT·
                                ;       RSPCNT·  - NUMBER·OF·WORDS·TO·EXPECT·IN·COMMAND·LINE·
                                ;
                                ;       OUTPUT:
                                ;       WWORDS·(DATA1, DATA2, DATA3, DATA4) DEPENDING·UPON·RSPCNT·
                                ;
                                ;       C-BIT·CLEAR, V-BIT·CLEAR·       NORMAL·RETURN·
                                ;       C-BIT·CLEAR, V-BIT·SET          <CR> RESPONSE·TO·PROMPT·
                                ;       C-BIT·SET, V-BIT·CLEAR          END·OF·MEMORY·OR·CONVERSION·ERROR·
                                ;
                                ;       REGISTERS·1, 4, 5 DESTROYED·
                                ;
                                ;
002764                          PDATA::
002764  026767  175204  175204          CMP·    MSTR2,MEND              ;UPPER·MEMORY·LIMIT·REACHED
002772· 101403                          BLOS·   10$                     ;NO, CONTINUE·
002774                                  CALL·   ENDMEM·                 ;END·OF·MEMORY·REACHED·
003000  000460                          BR      PDCX                    ;SET·CARRY AND·EXIT·
                                ;
003002· 016701  175166          10$:    MOV·    MSTR2,R1                ;PREPARE·TO·PRINT·ADDRESS·
003006  012705  000550'                 MOV·    #PRINT,R5               ;POINT·TO·PRINT·LINE
003012                                  CALL·   UNPK                    ;CONVERT·ADDRESS·
003016                                  CALL·   CONSOL·                 ;PRINT·OUT ADDRESS·
                                ;
003022· 012767  020040  177016          MOV·    #20040,GCMBLK+G.DPRM·   ;ERASE·CR·AND·LF·
003030  012767  020040  177012·         MOV·    #20040,GCMBLK+G.DPRM+2· ;ERASE·PROCESSOR·MNEMONIC·
003036                                  CALL·   GCONLY·                 ;PROMPT·
003042· 112767  000015  176776          MOVB·   #15,GCMBLK+G.DPRM·      ;RESTORE·CR·
003050  112767  000012· 176771          MOVB·   #12,GCMBLK+G.DPRM+1     ;RESTORE·LF·
003056                                  CALL·   FIND                    ;LOCATE·FIRST·DATA WORD·IN·COMMAND·LINE·
003062· 103424                          BCS·    PDVX                    ;<CR> RESPONSE, EXIT
                                ;
003064· 012705  000206'                 MOV·    #WWORDS,R5              ;WORDS·FROM·COMMAND·LINE·GO·INTO·THIS·TABLE·
003070  016704  175106                  MOV·    RSPCNT,R4               ;NUMBER·OF·WORDS·TO·EXPECT·
003074                          2$:     CALL·   PACK                    ;CONVERT·WORD·TO·BINARY·
003100  103003                          BCC·    3$                      ;OK, CONTINUE·
003102·                                 CALL·   ERR4                    ;'INVALID·NUMERIC·VALUE'
003106  000415                          BR      PDCX                    ;AND·EXIT·
003110  016725  174716          3$:     MOV·    BINWD,(R5)+             ;MOVE·WORD·TO·TABLE·
003114  005304                          DEC·    R4                      ;SUB·FROM·LOOP·COUNT·
003116  001414                          BEQ·    PDCCX·                  ;FINISHED·
003120                                  CALL·   FIND                    ;FIND·NEXT·WORD·
003124  103363                          BCC·    2$                      ;OK, CONTINUE·
```

```
 569 003126
 570 003132  000403                    CALL    ERR3                    ;'MISSING OPERAND'
 571                                    BR      PDCX
                                    ;
 572 003134  000241             PDVX:   CLC
 573 003136  000262                     SEV
 574 003140  000405                     BR      PDDX                    ;INDICATE <CR> RESPONSE
 575 003142  000242             PDCX:   CLV
 576 003144  000261                     SEC
 577 003146  000402                     BR      PDDX                    ;END OF MEMORY OR CONVERSION ERROR
 578 003150  000241             PDCCX:  CLC
 579 003152  000242                     CLV
 580 003154                     PDDX:   RETURN
```

```
                                        ;
                                        ;
                                        ;       COMMON PRINT ROUTINE
                                        ;       THIS ROUTINE CONVERTS VALUES IN THE FIELD 'WWORDS' TO
                                        ;       ASCII HEX (DEPENDING ON RSPCNT) AND PRINTS THEM (TT0)
                                        ;
                                        ;       INPUT:
                                        ;       #OUT FLAG       - WHEN LOOP OPTION IS IN EFFECT, PRINT 1 WORD ONLY
                                        ;       #ONCE FLAG      - CONTROL PRINTING OF 1 WORD WHEN #OUT IS SET
                                        ;       RSPCNT          - NUMBER OF WORDS TO PRINT (SEE 'BUFSET' RTN)
                                        ;       MSTR2           - MEMORY ADDRESS (SEE 'BUFSET')
                                        ;       WWORDS          - MEMORY WORDS (DATA1, DATA2, DATA3, DATA4)
                                        ;
                                        ;       REGISTERS 1, 3, 4, 5 DESTROYED
                                        ;
                                        ;
003156                                  PRDATA::
003156  032767  002000  174650                  BIT     #OUT,BASE       ;OUTPUT CONTROL ON
003164  001407                                  BEQ     10$             ;NO, SKIP OUTPUT CONTROL
003166  032767  000100  174640                  BIT     #ONCE,BASE      ;ONE LINE PRINTED
003174  001026                                  BNE     PRDX            ;YES, EXIT
003176  052767  000100  174630                  BIS     #ONCE,BASE
                                        ;
003204  016701  174764                  10$:    MOV     MSTR2,R1        ;FIRST CONVERT ADDRESS TO ASCII
003210  012705  000550'                         MOV     #PRINT,R5       ;POINT TO PRINT LINE
003214                                          CALL    UNPK            ;PERFORM CONVERSION
                                        ;
003220  012704  000206'                         MOV     #WWORDS,R4      ;POINT TO WORDS FOR PRINTING
003224  016703  174752                         MOV     RSPCNT,R3       ;NUMBER OF MEM WORDS TO PRINT
003230  062705  000002                 1$:     ADD     #2,R5           ;ADVANCE PRINT LINE POINTER
003234  012401                                 MOV     (R4)+,R1        ;LOAD ONE WORD
003236                                          CALL    UNPK            ;CONVERT IT TO ASCII
003242  005303                                 DEC     R3              ;DEC WORD COUNT
003244  001371                                 BNE     1$              ;REPEAT
003246                                          CALL    CONSOL          ;PRINT LINE
                                        ;
003252                                  PRDX:   RETURN
```

```
                                ;
                                ;
                                ;       GET COMMAND LINE FROM TERMINAL
                                ;
                                ;       OUTPUT:
                                ;       GCMBUF  - WORK AREA TO HOLD COMMAND LINE
                                ;       GCMLEN  - LENGTH OF LINE READ
                                ;       GCMPNT  - POINTER TO COMMAND LINE (SET TO POINT TO BEGINNING)
                                ;
                                ;       SEE 'FIND' SUBROUTINE FOR THE USE AND UPDATING OF THESE FIELDS.
                                ;
                                ;
003254                          GCONLY::
003254  012700  000040'                MOV     #GCMBUF,R0              ;POINT TO COMMAND LINE BUFFER
003260  012701  000051                 MOV     #41.,R1                 ;NUMBER OF WORDS IN BUFFER
003264  005020                 1$:     CLR     (R0)+                   ;CLEAR COMMAND LINE
003266  005301                         DEC     R1                      ;FINISHED?
003270  001375                         BNE     1$                      ;NO
                                ;
003272                                 GCML$   #GCMBLK
003306  016067  000146  174646        MOV     G.CMLD(R0),GCMLEN       ;SAVE LENGTH
003314  012767  000040' 174642        MOV     #GCMBUF,GCMPNT          ;INIT COMMAND LINE POINTER
003322  012701  000040'                MOV     #GCMBUF,R1              ;POINT R1 TO COMMAND LINE
003326                                 RETURN
```

```
                                ;
                                ;
                                ;       INTERRUPT SERVICE ROUTINE
                                ;       TRAP INTERRUPTS FROM HOR THROUGH VECTOR ADDRESS 274
                                ;       SET EVENT FLAG 3
                                ;       MICROCODE (MRP AND CP) DEBUGGING ROUTINES WILL READ CSR #2 AND
                                ;       DECODE THE INTERRUPT
                                ;
                                ;
003330                          BPTISR:
003330                                  SAVE    R0,R1,R2,R3,R4,R5
                                ;
003344  016705  174434                  MOV     TSKTCB,R5               ;LOAD MY TCB
003350  012700  000003                  MOV     #EFN,3,R0               ;EVENT FLAG TO BE SET
003354                                  CALL    $CEFI
003360  050011                          BIS     R0,(R1)                 ;SET LOCAL FLAG
003362                                  CALL    $DRDSE                  ;DECLARE SIGNIFICANT EVENT
                                ;
003366                                  RESTOR  R0,R1,R2,R3,R4,R5
003402  000002                          RTI
```

```
                                ;
                                ;
                                ;       DETECT LOOP INDICATOR IN COMMAND LINE
                                ;
                                ;
003404                          LOOPR::
003404                                  CALL    FIND                    ;LOCATE RESPONSE
003410  103437                         BCS     LOOPX                   ;NOTHING THERE, EXIT
003412  122711  000114                  CMPB    #'L,(R1)                ;LOOP INDICATOR
003416  001403                         BEQ     HANG                    ;OK, CONTINUE
003420                                  CALL    ERR11                   ;WRONG CHARACTER
003424  000431                         BR      LOOPX
                                ;
                                ;       GIVE DIRECTIONS FOR EXITING LOOP
                                ;       ISSUE QIO (ATTACH) FOR UNSOLICITED CHARACTER INTERRUPT
                                ;
003426  052767  000200  174400  HANG::  BIS     #ASTFLG,BASE            ;SET FLAG FOR QIO ISSUED
003434  052767  000004  174372          BIS     #LOOP,BASE              ;SET FLAG FOR LOOP
003442                                  CALL    STOP                    ;GIVE DIRECTIONS FOR STOPPING TEST
                                ;
003446                                  QIO$S   #IO.ATA,#LUN,TT,,,,,<#AST>
                                ;
003510                          LOOPX:
003510                                  RETURN
                                ;
                                ;
                                ;       HANG QIO FOR INTERRUPT FROM TERMINAL (BREAKPOINT RTNS)
                                ;
                                ;
003512                          HANG2::
003512  052767  000200  174314          BIS     #ASTFLG,BASE            ;SET FLAG FOR QIO ISSUED
003520                                  QIO$S   #IO.ATA,#LUN,TT,,,,,<#AST2>
003562                                  RETURN
                                ;
                                ;
                                ;       KILL AST (BY DETACH)
                                ;       DETACH IS ISSUED BECAUSE THE LOOP QIO ATTACH AND THE
                                ;       BREAKPOINT QIO ATTACH EACH HAS ITS OWN AST.
                                ;
                                ;
003564                          KILL::
003564  042767  000004  174242          BIC     #LOOP,BASE              ;CLEAR LOOP FLAG
003572  032767  000200  174234          BIT     #ASTFLG,BASE            ;HAS A QIO BEEN ISSUED?
003600  001423                         BEQ     1$                      ;NO, DO NOT DETACH
003602  042767  000200  174224          BIC     #ASTFLG,BASE            ;CLEAR QIO FLAG
003610                                  QIOW$S  #IO.DET,#LUN,TT
003650                          1$:     RETURN
```

```
714                                     ;
715                                     ;       AST FOR LOOP ON TEST QIO
716                                     ;
717 003652                              AST:
718 003652  012667  174310              MOV     (SP)+,ASTWRD            ;GET CHAR OFF STACK
719 003656  042767  000004  174150      BIC     #LOOP,BASE              ;CLEAR LOOP FLAG
720 003664                              ASTX$S
721                                     ;
722                                     ;       AST FOR BREAKPOINT QIO
723                                     ;
724 003672                              AST2:
725 003672  012667  174270              MOV     (SP)+,ASTWRD
726 003676                              SETF$S  #EFN.4                  ;SET EVENT FLAG FRO TERMINAL INTERRUPT
727                                     ;
728 003710                              ASTX$S
```

```
                                        ;       SCAN A TABLE FOR A VALID COMMAND/MNEMONIC.
                                        ;
                                        ;       INPUT:
                                        ;       R0 = NUMBER OF ENTRIES IN COMMAND TABLE.
                                        ;       R1 -> CHAR STRING IN GCML COMMAND LINE.
                                        ;       R2 -> TOP OF COMMAND TABLE.
                                        ;
                                        ;       OUTPUT:
                                        ;       R1 -> ROUTINE THAT GOVERNS THE COMMAND (IF MATCH WAS MADE)
                                        ;       R1 -> CHAR STRING IN COMMAND LINE (IF NO MATCH WAS MADE)
                                        ;       R0 = RELATIVE POSITION OF MATCHED ENTRY IN TABLE.
                                        ;
003716                                  SCAN::
003716  010346                                  MOV     R3,-(SP)                ;SAVE R3
003720  010046                                  MOV     R0,-(SP)                ;SAVE # ENTRIES
003722  010146                                  MOV     R1,-(SP)                ;SAVE POINTER TO BEGINNING OF STRING
                                        ;
003724  011601                          FNOUT1: MOV     (SP),R1                 ;POINT TO NON-BLANK IN COMMAND LINE
003726  012703  000002                          MOV     #2,R3                   ;NUMBER OF CHARS IN NON-BLANK FIELD
003732  122122                          FNIN1:  CMPB    (R1)+,(R2)+             ;DOES COMMAND LINE MATCH TABLE ENTRY
003734  001003                                  BNE     FNOUT2                  ;NO, TRY NEXT TABLE ENTRY
003736  005303                                  DEC     R3                      ;SUB FROM LOOP COUNT
003740  001374                                  BNE     FNIN1
003742  000411                                  BR      FNMTCH                  ;COMMAND FOUND IN TABLE
003744  060302                          FNOUT2: ADD     R3,R2                   ;ADD # UNCOMPARED CHARS TO POINTER
003746  005202                                  INC     R2                      ;THEN ADJUST TO NEXT TABLE ENTRY
003750  005300                                  DEC     R0                      ;SUB FROM OUTER LOOP COUNT
003752  001364                                  BNE     FNOUT1                  ;TRY AGAIN
003754  012601                                  MOV     (SP)+,R1                ;RESTORE R1
003756  012600                                  MOV     (SP)+,R0
003760  012603                                  MOV     (SP)+,R3
003762  000261                                  SEC                             ;COMMAND NOT IN TABLE
003764                                  RETURN
                                        ;
003766  010201                          FNMTCH: MOV     R2,R1                   ;POINT R1 AT RTN ADDR IN TABLE
003770  062706  000002                          ADD     #2,SP                   ;POINT TO INCOMING R0 ON STACK
003774  012602                                  MOV     (SP)+,R2                ;GET TOTAL # TABLE ENTRIES
003776  160002                                  SUB     R0,R2                   ;GET POSITION OF MATCHED ENTRY
004000  010200                                  MOV     R2,R0                   ;PUT IN R0 FOR RETURN
004002  012603                                  MOV     (SP)+,R3                ;RESTORE R3
004004  000241                                  CLC
004006                                  RETURN
```

```
                                ;       FIND THE NEXT NON-BLANK IN THE COMMAND BUFFER,
                                ;       THEN FIND THE LENGTH OF THE STRING THAT STARTS WITH THAT CHARACTER.
                                ;
                                ;       INPUT:
                                ;       GCMLEN  -  NUMBER OF UNPROCESSED BYTES IN COMMAND LINE
                                ;       GCMPNT  - ADDR OF NEXT UNPROCESSED POSITION IN COMMAND LINE
                                ;
                                ;       OUTPUT:
                                ;       R1 -> STRING, R0 = LENGTH OF STRING
                                ;       GCMLEN, GCMPNT UPDATED FOR NEXT ENTRY INTO THIS ROUTINE
                                ;
                                ;       THIS ROUTINE IS DESIGNED TO BE ENTERED A NUMBER OF TIMES
                                ;       IN THE PARSING OF A COMMAND LINE. THE FIELDS GCMLEN AND
                                ;       GCMPNT ARE REFRESHED WHEN A NEW COMMAND LINE IS READ
                                ;       (SEE THE SUBROUTINE 'GCONLY').
                                ;
                                ;
004010                          FIND::
004010  010246                          MOV     R2,-(SP)                ;SAVE R2
004012  016701  174144                  MOV     GCMLEN,R1               ;# BYTES REMAINING IN COMMAND BUFFER
004016  001440                          BEQ     FSECX                   ;THERE ARE NONE
004020  016702  174140                  MOV     GCMPNT,R2               ;LOAD CURRENT POINTER
004024  122712  000040          1$:     CMPB    #40,(R2)                ;LOOK FOR A BLANK
004030  001403                          BEQ     10$                     ;OK, BUMP TO NEXT CHAR
004032  122712  000054                  CMPB    #',,(R2)                ;COMMA IN COMMAND LINE
004036  001004                          BNE     2$                      ;TREAT COMMA AS BLANK
004040  005202                  10$:    INC     R2                      ;BUMP POINTER
004042  005301                          DEC     R1                      ;SUB FROM REMAINING LENGTH
004044  001367                          BNE     1$
004046  000424                          BR      FSECX                   ;NO NON-BLANK FOUND
                                ;
004050  010246                  2$:     MOV     R2,-(SP)                ;TEMP SAVE POINTER TO BEGINNING OF STRING
004052  005000                          CLR     R0                      ;CLEAR CHAR COUNT
004054  122712  000040          3$:     CMPB    #40,(R2)                ;LOOK FOR A BLANK
004060  001407                          BEQ     4$                      ;FOUND END OF STRING
004062  122712  000054                  CMPB    #',,(R2)                ;TREAT COMMAS AS BLANKS
004066  001404                          BEQ     4$
004070  005202                          INC     R2                      ;BUMP POINTER
004072  005200                          INC     R0                      ;BUMP CHAR COUNT
004074  005301                          DEC     R1                      ;SUB FROM BYTES REMAINING
004076  001366                          BNE     3$
                                ;
004100  010267  174060          4$:     MOV     R2,GCMPNT               ;SAVE POINTER FOR NEXT TIME
004104  010167  174052                  MOV     R1,GCMLEN               ;SAVE BYTES REMAINING FOR NEXT TIME
004110  012601                          MOV     (SP)+,R1                ;POINTER TO BEGINNING OF STRING
004112  012602                          MOV     (SP)+,R2                ;RESTORE R2
004114  000241                          CLC
004116                                  RETURN
                                ;
004120  012602                  FSECX:  MOV     (SP)+,R2                ;RESTORE R2
004122  000261                          SEC
004124                                  RETURN
```

```
                                ;
                                ;
                                ;       CONVERT AN ASCII HEX VALUE FROM THE COMMAND LINE INTO BINARY.
                                ;       LEGAL STRINGS CONTAIN FROM 1 TO 4 CHARACTERS.
                                ;
                                ;       INPUT:
                                ;       R0 = NUMBER OF CHARACTERS IN ASCII STRING.
                                ;       R1 -> STRING
                                ;
                                ;       OUTPUT:
                                ;       THE FIELD 'BINWD' CONTAINS THE CONVERTED VALUE.
                                ;
                                ;
004126                          PACK::
004126  005067  173700                  CLR     BINWD                   ;CLEAR CONVERTED VALUE FIELD.
004132  005046                          CLR     -(SP)                   ;CLEAR FOR COND CODE INDICATOR
004134                                  SAVE    R0,R1,R2,R3,R4,R5
                                ;
                                ;       DETERMINE THE CONVERSION FACTOR (POWER OF 16) FOR
                                ;       THE LEFTMOST ASCII CHARACTER.
                                ;
004150  022700  000004                  CMP     #4,R0                   ;UPPER LIMIT FOR HEX DIGITS.
004154  002455                          BLT     PSECX                   ;ERROR EXIT.
004156  010002                          MOV     R0,R2                   ;NUMBER OF CHARS CONTROLS LOOP
004160  022700  000004                  CMP     #4,R0                   ;4 CHARS ?
004164  001003                          BNE     1$
004166  012700  010000                  MOV     #4096.,R0               ;HEX CONVERSION FACTOR FOR HIGH ORDER CHAR.
004172  000416                          BR      4$                      ;ENTER LOOP.
004174  022700  000003          1$:     CMP     #3,R0                   ;3 CHARS ?
004200  001003                          BNE     2$
004202  012700  000400                  MOV     #256.,R0                ;CONVERSION FACTOR FOR HIGH ORDER CHAR.
004206  000410                          BR      4$                      ;ENTER LOOP.
004210  022700  000002          2$:     CMP     #2,R0                   ;2 CHARS ?
004214  001003                          BNE     3$
004216  012700  000020                  MOV     #16.,R0                 ;CONVERSION FACTOR FOR HIGH ORDER CHAR.
004222  000402                          BR      4$
004224  012700  000001          3$:     MOV     #1,R0                   ;1 CHAR.
                                ;
                                ;       MULTIPLY EACH CHARACTER'S VALUE BY ITS CONVERSION
                                ;       FACTOR. THE CONVERSION FACTOR IS REDUCED BY A POWER
                                ;       OF 16 AS THE ASCII STRING IS SCANNED FROM LEFT TO
                                ;       RIGHT.
                                ;
004230  010105                  4$:     MOV     R1,R5                   ;MOVE INPUT POINTER TO R5
004232  112503                  HLOOP:  MOVB    (R5)+,R3                ;GET ASCII VALUE INTO A REG.
004234  012704  000276'                 MOV     #TRTBL,R4               ;POINT TO TRANSLATE TABLE.
004240  060304                          ADD     R3,R4                   ;ADD VALUE OF CHARACTER
004242  111401                          MOVB    (R4),R1                 ;MOVE BINARY VALUE TO A REG.
004244  022704  000356'                 CMP     #TRTBL+60,R4            ;WAS INPUT CHAR ZERO
004250  001402                          BEQ     1$                      ;YES, THIS IS OK.
004252  105701                          TSTB    R1                      ;WAS TABLE POSITION EMPTY.
004254  001415                          BEQ     PSECX                   ;YES, TRANSLATION UNSUCCESSFUL
004256  010046                  1$:     MOV     R0,-(SP)                ;NSAVE FOR DIVISION LATER.
004260                                  CALL    $MUL                    ;MULT BY 16 TO SOME POWER.
004264  060167  173542                  ADD     R1,BINWD                ;ACCUM CONVERTED VALUE.
004270  012600                          MOV     (SP)+,R0                ;RELOAD FACTOR.
004272  012701  000020                  MOV     #16.,R1                 ;LOAD DIVISOR.
```

```
 883 004276                                     CALL    $DIV                    ;REDUCE FACTOR
 884 004302  005302                             DEC     R2                      ;SUB FROM LOOP COUNT
 885 004304  001352                             BNE     HLOOP
 886 004306  000403                             BR      PCLCX                   ;EXIT
 887                                    ;
 888 004310  012766  177777  000014  PSECX:     MOV     #-1,12.(SP)             ;SET COND CODE INDICATOR
 889 004316                            PCLCX:   RESTOR  R0,R1,R2,R3,R4,R5
 890                                    ;
 891 004332  005726                             TST     (SP)+                   ;GET COND CODE INDICATOR
 892 004334  002402                             BLT     1$                      ;DO SEC
 893 004336  000241                             CLC
 894 004340  000401                             BR      PACKX                   ;AND RETURN
 895 004342  000261                     1$:     SEC
 896 004344                             PACKX:  RETURN
```

```
                                ;
                                ;       CONVERT A VALUE FROM BINARY TO PRINTABLE FORM.
                                ;       R1 = WORD TO BE CONVERTED
                                ;       R5 -> PRINT LINE
                                ;
                                ;
004346                          UNPK::
004346                                  SAVE    R0,R1,R2,R3,R4
                                ;
004360  062705  000004                  ADD     #4,R5                   ;DO LAST CHAR FIRST
004364  012702  000004                  MOV     #4,R2                   ;NUMBER OF HEX DIGITS FOR A WORD
004370  010100                          MOV     R1,R0                   ;SUBRTN EXPECTS DIVIDEND IN R0
004372  012701  000020          1$:     MOV     #16.,R1                 ;LOAD DIVIDOR
004376                                  CALL    $DIV
004402  012703  000475'                 MOV     #TRTBL2,R3              ;POINT TO TRANSLATE TABLE
004406  060103                          ADD     R1,R3                   ;ADD 4 BIT VALUE
004410  111345                          MOVB    (R3),-(R5)              ;MOVE CHAR TO PRINT LINE
004412  005302                          DEC     R2                      ;DEC INNER LOOP COUNT
004414  001366                          BNE     1$
004416  062705  000005                  ADD     #5,R5                   ;BUMP PRINT LINE POINTER
                                ;
004422                                  RESTOR  R0,R1,R2,R3,R4
004434                                  RETURN
```

```
                                ;
                                ;
                                ;       WRITE A PRINT LINE TO TT0
                                ;
                                ;
004436                          CONSOL::
004436                                  SAVE    R0,R1
                                ;
004442  012700  000120                  MOV     #80.,R0                 ;PRINT BUFFER BYTE COUNT
004446  012701  000666'                  MOV     #PRINT+78.,R1           ;POINT PAST END OF BUFFER
004452  122741  000040          1$:     CMPB    #40.,-(R1)              ;LOOK FOR A NON-BLANK
004456  001003                          BNE     2$                      ;OK, WRITE LINE
004460  005300                          DEC     R0                      ;DEC CHAR COUNT
004462  001373                          BNE     1$
004464  000440                          BR      ABEND2                  ;NO NON-BLANKS?
                                ;
004466                          2$:     QIOW$S  #IO.WVB,#LUN.TT,#EFN.1,,,#STAT,,<#PRINT-2,R0>,ABEND2
                                ;
                                ;
004544  012701  000550'                  MOV     #PRINT,R1               ;POINT TO STRING
004550  112721  000040          4$:     MOVB    #40.,(R1)+              ;CLEAR LINE TO BLANKS
004554  005300                          DEC     R0                      ;DEC LOOP COUNT
004556  001374                          BNE     4$
                                ;
004560                                  RESTOR  R0,R1
004564                                  RETURN
                                ;
004566                          ABEND2: ABRT$S  #MYSELF
```

```
                                ;
                                ;
                                ;       WRITE TO TT0 AND PROMPT
                                ;
                                ;       REGISTERS 0, 1 ,2 DESTROYED
                                ;
                                ;
004620  005267  173204        ENDTST:: INC     ERWORD                  ;INC MESSAGE INDEX
004624  005267  173200        TIME::   INC     ERWORD
004630  005267  173174        ENFILE:: INC     ERWORD
004634  005267  173170        ENDMEM:: INC     ERWORD
004640  005267  173164        STOP::   INC     ERWORD
004644  005267  173160        OUT1::   INC     ERWORD
004650  005267  173154        ERR12::  INC     ERWORD
004654  005267  173150        ERR11::  INC     ERWORD
004660  005267  173144        ERR10::  INC     ERWORD
004664  005267  173140        ERR9::   INC     ERWORD
004670  005267  173134        ERR8::   INC     ERWORD
004674  005267  173130        ERR7::   INC     ERWORD
004700  005267  173124        ERR6::   INC     ERWORD
004704  005267  173120        ERR5::   INC     ERWORD
004710  005267  173114        ERR4::   INC     ERWORD
004714  005267  173110        ERR3::   INC     ERWORD
004720  005267  173104        ERR2::   INC     ERWORD
004724  005267  173100        ERR1::   INC     ERWORD
                                ;
                                ;       USE THE INDEX ERWORD TO COUNT UP FROM THE BOTTOM OF THE
                                ;       MESSAGE TABLE, FIND THE END OF THE MESSAGE FIRST, THEN
                                ;       THE BEGINNING, THEN GET THE LENGTH.
                                ;
004730  016702  173074                MOV     ERWORD,R2               ;LOAD LOOP COUNT
004734  012701  001665'              MOV     #ASCIZ,R1               ;POINT TO END OF MESSAGE TABLE
004740  105741                1$:     TSTB    -(R1)                   ;LOOK FOR END OF MESSAGE
004742  001376                        BNE     1$
004744  005302                        DEC     R2                      ;LOOP COUNT
004746  001374                        BNE     1$                      ;BACK UP ANOTHER MESSAGE
004750  010100                        MOV     R1,R0                   ;SAVE POINTER TO END OF MESSAGE
004752  105741                2$:     TSTB    -(R1)                   ;BACK UP TO BEGINNING OF MESSAGE
004754  001376                        BNE     2$
004756  005201                        INC     R1                      ;BUMP TO FIRST CHAR OF MESSAGE
004760  160100                        SUB     R1,R0                   ;R0 NOW = MESSAGE LENGTH
                                ;
004762                                  QIOW$S  #IO.WVB,#LUN,TT,#EFN,,,#STAT,,<R1,R0>,ABEND
                                ;
005036                                  CLEF$S  #EFN,1
005050  105767  172750                TSTB    STAT                    ;GOOD RETURN
005054  003403                        BLE     ABEND                   ;NO
                                ;
                                ;       CLEAR ERROR INDEX AND EXIT
                                ;
005056  005067  172746                CLR     ERWORD
005062                                  RETURN                          ;AND RETURN
                                ;
005064                          ABEND:  ABRT$S  #MYSELF
002174'                                 .END    START
```

```
ABEND   005064R
ABEND2  004566R
ALUCKE= 040000
ALUOE = 004000
APLACE  000036RG
ASCIZ   001665RG
AST     003652R
ASTFLG= 000200 G
ASTWRD  000166RG
AST2    003672R
A01   = 010000
BASE    000034RG
BCE   = ****** GX
BINWD   000032RG
BITVAL= 000000
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BLHIGH  000256PG
BLLOW   000260RG
BPTISR  003330R
BREAK = 004000 G
BUFCX2  002750R
BUFSET  002612RG
BUFSM   002630RG
BUFS2   002574RG
BUFS3   002556RG
BUFS4   002540RG
BUFXX2  002762R
BYTE0 = 000000
BYTE1 = 000001
BYTE10= 000012
BYTE11= 000013
BYTE12= 000014
BYTE13= 000015
BYTE14= 000016
BYTE15= 000017
BYTE16= 000020
BYTE17= 000021
BYTE18= 000022
BYTE19= 000023
BYTE2 = 000002
BYTE20= 000024
BYTE21= 000025
BYTE22= 000026
BYTE23= 000027
BYTE24= 000030
BYTE25= 000031
BYTE26= 000032
BYTE27= 000033
BYTE28= 000034
BYTE29= 000035
BYTE3 = 000003
BYTE30= 000036
BYTE31= 000037
BYTE32= 000040
BYTE33= 000041
BYTE34= 000042
BYTE35= 000043
BYTE36= 000044
BYTE37= 000045
BYTE38= 000046
BYTE39= 000047
BYTE4 = 000004
BYTE40= 000050
BYTE41= 000051
BYTE42= 000052
BYTE43= 000053
BYTE44= 000054
BYTE45= 000055
BYTE46= 000056
BYTE47= 000057
BYTE48= 000060
BYTE49= 000061
BYTE5 = 000005
BYTE50= 000062
BYTE51= 000063
BYTE52= 000064
BYTE53= 000065
BYTE54= 000066
BYTE55= 000067
BYTE56= 000070
BYTE57= 000071
BYTE58= 000072
BYTE59= 000073
BYTE6 = 000006
BYTE60= 000074
BYTE61= 000075
BYTE62= 000076
BYTE63= 000077
BYTE64= 000100
BYTE65= 000101
BYTE66= 000102
BYTE67= 000103
BYTE68= 000104
BYTE69= 000105
BYTE7 = 000007
BYTE70= 000106
BYTE71= 000107
BYTE72= 000110
BYTE73= 000111
BYTE74= 000112
BYTE75= 000113
BYTE76= 000114
BYTE77= 000115
BYTE78= 000116
BYTE79= 000117
BYTE8 = 000010
BYTE80= 000120
BYTE81= 000121
BYTE82= 000122
BYTE83= 000123
BYTE84= 000124
BYTE85= 000125
BYTE86= 000126
BYTE87= 000127
BYTE88= 000130
BYTE89= 000131
BYTE9 = 000011
BYTE90= 000132
BYTE91= 000133
BYTE92= 000134
BYTE93= 000135
BYTE94= 000136
BYTE95= 000137
BYTE96= 000140
BYTE97= 000141
BYTE98= 000142
BYTE99= 000143
BYTVAL= 000144
CBKALL= 001000
CBKCLK= 000400
CDHIGH  000232RG
CDLOW   000234RG
CL      002450RG
CMILUN= 000002
CNOBRE= 100000
COMXX   002334RG
CONSOL  004436RG
CP    = ****** GX
CPCCEN= 010000
CPREAD= 040000
CPWRTE= 020000
CSADRD= 000004
CSEQCI= 100000
CSHIGH  000226RG
CSLOW   000230RG
CSOE  = 000040
CSR1    002410RG
CSWRTE= 000100
DATA1   000206RG
DATA2   000210RG
DATA3   000212RG
DATA4   000214RG
DBR.RD= 000001
DB$CPP= 001457
DB$SPT= 000026
DB$TPC= 000023
DISPGS= 100000
DMAAWR= 000005
DMARRD= 000003
DMARWR= 000004
EFBUF   000014RG
EFN.1 = 000001 G
EFN.2 = 000002 G
EFN.3 = 000003 G
EFN.33= 000041 G
EFN.4 = 000004 G
ENBR  = 010000
ENDMEM  004634RG
ENDTST  004620RG
ENFILE  004630RG
ERR1    004724RG
ERR10   004660RG
ERR11   004654RG
ERR12   004650RG
ERR2    004720RG
ERR3    004714RG
ERR4    004710RG
ERR5    004704RG
ERR6    004700RG
ERR7    004674RG
ERR8    004670RG
ERR9    004664RG
ERWORD  000030RG
EXIT    002370R
FAHIGH  000242RG
FALOW   000244RG
FD.CCL= ****** GX
FD.REC= ****** GX
FD.TTY= ****** GX
FIND    004010RG
FNIN1   003732R
FNMTCH  003766R
FNOUT1  003724R
FNOUT2  003744R
FNUM  = 000006
FSECX   004120R
FTBL    000516R
F.ACTL= 000076
F.ALOC= 000040
F.BBFS= 000062
F.BDB = 000070
F.BGBC= 000057
F.BKDN= 000026
F.BKDS= 000020
F.BKEF= 000050
F.BKP1= 000051
F.BKST= 000024
F.BKVB= 000064
F.CHR = 000075
F.CNTG= 000034
F.DFNB= 000046
F.DSPT= 000044
F.DVNM= 000134
F.EFBK= 000010
F.EFN = 000050
F.EOBB= 000032
F.ERR = 000052
F.FACC= 000043
F.FFBY= 000014
F.FNAM= 000110
F.FNB = 000102
F.FTYP= 000116
F.FVER= 000120
F.HIBK= 000004
F.LUN = 000042
F.MBCT= 000054
F.MBC1= 000055
F.MBFG= 000056
F.NRBD= 000024
F.NREC= 000030
F.OVBS= 000030
F.RACC= 000016
F.RATT= 000001
F.RCNM= 000034
F.RCTL= 000017
F.RSIZ= 000002
F.RTYP= 000000
F.SEQN= 000100
F.SPDV= 000072
F.SPUN= 000074
F.STBK= 000036
F.UNIT= 000136
F.URBD= 000020
F.VBN = 000064
F.VBSZ= 000060
GCMBLK  001666RG
GCMBUF  000040RG
GCMLEN  000162RG
GCMPNT  000164RG
GCONLY  003254RG
GE.BIF= 177775
GE.CLO= 000004
GE.COM= 000001
GE.CON= 000020
GE.EOF= 177766
GE.IND= 000002
GE.IOR= 177777
GE.LC = 000010
GE.MDE= 177774
GE.OPR= 177776
GE.RBG= 177730
GE.SIZ= 000040
G.CMLD= 000146
G.DPRM= 000160 G
G.ERR = 000140
G.ISIZ= 000020
G.LRDL= 000060
G.MODE= 000144
G.RSDS= 000142
G.SIZE= 000224
```

```
HANG    003426RG    OLDVEC  000006RG    Q$LBP = 000001    S$S1  = 000010    T$RSET= 040000
HANG2   003512RG    ONCE  = 000100 G    Q$LDCD= 000003    S$S2  = 000014    T$SC  = 000022
HLHIGH  000252RG    OUT   = 002000 G    Q$LDMD= 000004    S.BFHD= 000020    T$SCLK= 020000
HLLOW   000254RG    OUTi    004644RG    Q$LDPP= 002000    S.FATT= 000016    T$SEG1= 000000
HLOOP   004232R     PACK    004126RG    Q$LHP = 010000    S.FDB = 000140    T$SEG2= 000001
INCVAL  000200RG    PACKX   004344R     Q$MNC = 140000    S.FNAM= 000006    T$SEG3= 000002
IO.ATA= ****** G    PAR$$$= 000027      Q$MR  = 000052    S.FNB = 000036    T$SO  = 000001
IO.DET= ****** G    PCLCX   004316R     Q$MRP = 000040    S.FNBW= 000017    T$UBUS= 100000
IO.RVB= ****** G    PDATA   002764RG    Q$MRP2= 000240    S.FNTY= 000004    T$1CLK= 000400
IO.WVB= ****** G    PDCCX   003150R     Q$MSC = 040000    S.FTYP= 000002    T$8BEN= 000020
KILL    003564RG    PDCX    003142R     Q$MSET= 000004    S.NFEN= 000020    UBD.IN= 000020
LHHIGH  000246RG    PDDX    003154R     Q$MSP = 100000    TD$CTR= 176370    UNPK    004346RG
LHLOW   000250RG    PDVX    003134R     Q$NCLK= 176000    TD$CTW= 176360    UPLIM   000204RG
LOADER  000010R     PLB   = 000010      Q$PP  = 000100    TD$INL= 004000    WORD0 = 000000
LOC.EN= 000100      PLC   = 000020      Q$PPSW= 000320    TD$MEM= 000270    WORD1 = 000002
LOC.WA= 040000      PLD   = 000030      Q$PP2 = 000300    TD$OAR= 176344    WORD10= 000024
LOC.WB= 100000      PLRWR = 000200      Q$QHLT= 000013    TD$OTR= 176346    WORD11= 000026
LOOP  = 000004 G    PLR.EN= 000200      Q$QL  = 000043    TD$QRD= 000274    WORD12= 000030
LOOPR   003404RG    PPCR  = ****** GX   Q$QLA = 000053    TD$SW = 176376    WORD13= 000032
LOOPX   003510R     PPS   = ****** GX   Q$QLB = 000054    TD$TAR= 176372    WORD14= 000034
LUN.TT= 000001 G    PRDATA  003156RG    Q$QLR = 000001    TD$TAW= 176362    WORD15= 000036
MAREN1= 000001      PRDX    003252R     Q$QW  = 000042    TD$TDR= 176374    WORD16= 000040
MAREN2= 004000      PRINT   000550RG    Q$RDCD= 000005    TD$TDW= 176364    WORD17= 000042
MARLOD= 010000      PSECX   004310R     Q$RDMD= 000006    TIME    004624RG  WORD18= 000044
MAROUT= 000002      QR$CR1= 176420      Q$REBK= 001000    TRTBL   000276RG  WORD19= 000046
MAR.LO= 002000      QR$CR2= 176422      Q$RNC = 006000    TRTBL2  000475RG  WORD2 = 000004
MAR.OU= 000040      QR$LBR= 176424      Q$RSC = 004000    TSKTCB  000004RG  WORD20= 000050
MBKALL= 001000      QXHIGH  000236RG    Q$RSET= 000010    T$AD  = 000020    WORD21= 000052
MBKCLK= 000400      QXLOW   000240RG    Q$SM  = 100000    T$BA  = 000002    WORD22= 000054
MDHIGH  000222RG    Q$ATTN= 000100      Q$SP  = 000120    T$BD  = 000010    WORD23= 000056
MDLOW   000224RG    Q$BCL = 000001      Q$SP2 = 000340    T$BSO = 100000    WORD24= 000060
MEND    000176RG    Q$CCCP= 000040      RESEL   002272R   T$BT  = 000020    WORD25= 000062
MMADRD= 000100      Q$CHB = 000400      RGQ.EN= 000200    T$BTAR= 000030    WORD26= 000064
MMHIGH  000216RG    Q$CHRL= 000200      RGQ.VA= 020000    T$BTD = 002000    WORD27= 000066
MMLEFT= 000002      Q$CLR = 000040      RP    = 001000 G  T$CD  = 000100    WORD28= 000070
MMLOW   000220RG    Q$CNC = 030000      RSPCNT  000202RG  T$CLK = 002000    WORD29= 000072
MMOE  = 000004      Q$CP  = 000060      RTNPT   000170RG  T$DISK= 000200    WORD3 = 000006
MMWRTE= 000010      Q$CPCC= 000010      SCAN    003716RG  T$DRD = 000004    WORD30= 000074
MNOBRE= 100000      Q$CP2 = 000260      SDHIGH  000272RG  T$EMEM= 010000    WORD31= 000076
MREN1 = 000001      Q$CSC = 010000      SDLOW   000274RG  T$FSAA= 000000    WORD32= 000100
MREN2 = 020000      Q$CSEL= 000360      SEQ.CI= 000010    T$FSAB= 000004    WORD33= 000102
MRP   = ****** GX   Q$CSET= 000002      SET     002644R   T$FSAC= 000014    WORD34= 000104
MSTRT   000172RG    Q$CSP = 020000      SPCR  = ****** GX T$FSB2= 000010    WORD35= 000106
MSTR2   000174RG    Q$DMA = 000001      SPS   = ****** GX T$IB  = 000026    WORD36= 000110
MSYN  = 000040      Q$ENBK= 040000      SQHIGH  000262RG  T$IBAR= 000024    WORD37= 000112
MYSELF  000000RG    Q$ENOP= 020000      SQLOW   000264RG  T$IBE = 020000    WORD38= 000114
N     = 000144      Q$FAL = 004000      SRHIGH  000266RG  T$IBF = 040000    WORD39= 000116
NEQLB = 010000 G    Q$FC  = 000045      SRLOW   000270RG  T$ICD = 000040    WORD4 = 000010
N.DID = 000024      Q$FO  = 000044      START   002174R   T$MODE= 004000    WORD40= 000120
N.DVNM= 000032      Q$FP  = 000046      STAT    000024RG  T$OB  = 000036    WORD41= 000122
N.FID = 000000      Q$HBF = 000002      STOP    004640RG  T$OBE = 004000    WORD42= 000124
N.FNAM= 000006      Q$ICP = 000006      S$CLR = 000000    T$OBF = 010000    WORD43= 000126
N.FTYP= 000014      Q$IHB = 000003      S$LA  = 000001    T$OBRA= 000034    WORD44= 000130
N.FVER= 000016      Q$IHRL= 000002      S$QB  = 000005    T$OBWA= 000032    WORD45= 000132
N.NEXT= 000022      Q$IMRP= 000007      S$QR  = 000006    T$OUTA= 100000    WORD46= 000134
N.STAT= 000020      Q$LBD = 001000      S$QX  = 000004    T$RBDO= 000200    WORD47= 000136
N.UNIT= 000034      Q$LBDP= 001001      S$SR  = 000007    T$RNB = 000040    WORD48= 000140
```

SYMBOL·TABLE·


```
WORD49=·000142·          WORD62=·000174          WORD76=·000230          WORD9 =·000022·          $CEFI·=·****** GX·
WORD5 =·000012·          WORD63=·000176          WORD77=·000232·         WORD90=·000264           $DIV  =·****** GX·
WORD50=·000144           WORD64=·000200          WORD78=·000234          WORD91=·000266           $DRDSE=·****** GX·
WORD51=·000146           WORD65=·000202·         WORD79=·000236          WORD92=·000270           $MUL  =·****** GX·
WORD52=·000150           WORD66=·000204          WORD8 =·000020          WORD93=·000272·          $TKTCB=·****** GX·
WORD53=·000152·          WORD67=·000206          WORD80=·000240          WORD94=·000274           $$$   =·002046R·
WORD54=·000154           WORD68=·000210          WORD81=·000242·         WORD95=·000276           $$$ARG=·000002
WORD55=·00015E           WORD69=·000212·         WORD82=·000244          WORD96=·000300           $$$T1 =·000067
WORD56=·000160           WORD7 =·000016          WORD83=·000246          WORD97=·000302·          $$$T2·=·000027
WORD57=·000162·          WORD70=·000214          WORD84=·000250          WORD98=·000304           .FSRCB=·****** G·
WORD58=·000164           WORD71=·000216          WORD85=·000252·         WORD99=·000306           .GCML1=·****** G·
WORD59=·000166           WORD72=·000220          WORD86=·000254          WRDVAL=·000310           ...PC1=·001666R·
WORD6 =·000014           WORD73=·000222·         WORD87=·000256          WWORDS·  000206R·        ...PC2=·002070R·
WORD60=·000170           WORD74=·000224          WORD88=·000260          XTREAD=·001000           ...PC3=·001666R·
WORD61=·000172·          WORD75=·000226          WORD89=·000262·         XTWRTE=·000400           ...TPC=·000020

. ABS.  000000     000
        005116     001
$$FSR1  001020     002·
ERRORS·DETECTED:  0

VIRTUAL·MEMORY·USED:  7917 WORDS· (·31 PAGES)
DYNAMIC·MEMORY:  9140 WORDS· ( 35 PAGES)
ELAPSED·TIME:  00:01:36
wMAIN›QMAIN/-SP=[20,1]IM›[20,1]QMAIN·
```

```
                                        .TITLE.GRSUB
000000                                  .PSECT.CPSUB
                                ;
                                ;       HARDWARE.QUERY.RESOLVER.'MANUAL' DEBUGGING AIDS.
                                ;       *******   PROTOTYPE.VERSION.  *******
                                ;
                                ;       CP SUBROUTINES.
                                ;
                                ;       CONTROL.STORE.SEQUENCING.(CONTROL.PROCESSOR)
                                ;       JUMP.TO.START.ADDRESS.(VIA.BRANCH.REGISTER)
                                ;
                                ;       INPUT:
                                ;       2(SP)   START.ADDRESS.
                                ;
                                ;       OUTPUT:
                                ;       CP SEQUENCER SET.TO.START.ADDRESS.
                                ;
000000                          CPSUB:
000000                          SEQCS::
                                ;
                                ;       STOP.CP.AND.MRP.CLOCKS
                                ;
000000  005046                          CLR.    -(SP)                   ;CLEAR.NOTHING.
000002  012746  176000                  MOV.    #Q$NCLK,-(SP)           ;SET.NO.CLOCKS.
000006                                  CALL.   CSR1
                                ;
                                ;       GO TO.LOAD.MODE.(AND.CLEAR.EVERYTHING.ELSE)
                                ;
000012  005067  176422                  CLR.    QR$CR2.                 ;DO.IT
                                ;
                                ;       SELECT.BRANCH.REGISTER IN.CP.
                                ;
000016  005046                          CLR.    -(SP)                   ;DEFINE.BR REG.SELECT.
000020                                  CALL.   CPCRA.                  ;WRITE.CR.(WITHOUT.AUTOMATIC.BR.INHIBIT)
                                ;
                                ;       CLOCK.CP.ONCE.
                                ;
000024  012746  030000                  MOV.    #Q$CNC,-(SP)            ;CLEAR.CP.NO.CLOCKS.
000030  012746  020000                  MOV.    #Q$CSP,-(SP)            ;SET.CP.SINGLE.CLOCK
000034                                  CALL.   CSR1
                                ;
                                ;       MOVE.INSTRUCTION.AND.ADDRESS.TO.BRANCH.REGISTER.VIA.LOAD.BUS.
                                ;       ISSUE.CLOCK.TO.PUT.START.ADDRESS CONTENTS.AT.INPUTS.TO.PLR.
                                ;
000040  052766  020000  000002          BIS.    #BIT13,2(SP)            ;OR.JMP.INSRUCTION.(02).INTO.HIGH.ORDER.NIBBLE.
000046  016646  000002                  MOV.    2(SP),-(SP)             ;COPY.ARGUMENT.INTO.STACK.FOR.LBCP.CALL.
000052                                  CALL.   LBCP                    ;DO.MOVE.- ISSUE.CLOCK.
                                ;
000056  011666  000002                  MOV.    (SP),2(SP)              ;MOVE.RETURN.ADDRESS.DOWN.STACK.
000062  005726                          TST.    (SP)+                   ;POINT.TO.RETURN.ADDRESS.
000064                                  RETURN.
```

```
                                ;
                                ;       CONTROL·STORE·SEQUENCING·(CONTROL·PROCESSOR)
                                ;       SET·START·ADDRESS·FOR·WRITE·
                                ;
                                ;       INPUT:
                                ;       2(SP)   START·ADDRESS·
                                ;
                                ;       OUTPUT·
                                ;       CP SEQUENCER SET·TO·START·ADDRESS·
                                ;
000066                          WRTCS::
                                ;
                                ;       STOP·CP·AND·MRP·CLOCKS
                                ;
000066  005046                  CLR·    -(SP)                   ;CLEAR·NOTHING·
000070  012746  176000          MOV·    #Q$NCLK,-(SP)           ;SET·NO·CLOCKS·
000074                          CALL·   CSR1
                                ;
                                ;       GO TO·LOAD·MODE·(AND·CLEAR·EVERYTHING·ELSE)
                                ;
000100  005067  176422          CLR·    QR$CR2·                 ;DO·IT
                                ;
                                ;       SELECT·BRANCH·REGISTER IN·CP·
                                ;
000104  005046                  CLR·    -(SP)                   ;DEFINE·BR REG·SELECT·
000106                          CALL·   CPCRA·                  ;WRITE·CR·(WITHOUT·AUTOMATIC·BR· INHIBIT)
                                ;
                                ;       CLOCK·CP·ONCE·
                                ;
000112· 012746  030000·         MOV·    #Q$CNC,-(SP)            ;CLEAR·CP·NO·CLOCKS·
000116  012746  020000·         MOV·    #Q$CSP,-(SP)            ;SET·CP·SINGLE·CLOCK
000122·                         CALL·   CSR1
                                ;
                                ;       MOVE·INSTRUCTION·AND·ADDRESS·TO·BRANCH·REGISTER·VIA·LOAD·BUS·
                                ;
000126  052766  020000  000002· BIS·    #BIT13,2(SP)            ;OR·JMP·INSRUCTION·(02) INTO·HIGH·ORDER·NIBBLE·
000134  016646  000002          MOV·    2(SP),-(SP)             ;COPY·ARGUMENT·INTO·STACK·FOR·LBCP·CALL·
000140                          CALL·   LBCP                    ;DO·MOVE·- FIRST·CLOCK·
                                ;
000144                          RETURN·
```

```
                                        ;
                                        ;       DATA TRANSFER
                                        ;       LOD BUS REGISTER TO A DESTINATION ON THE CP BUS
                                        ;
                                        ;       INPUT:
                                        ;       2(SP)   DATA FOR PRE-SELECTED CP DESTINATION
                                        ;
000146                                  LBCP::
000146  016667  000002  176424                  MOV     2(SP),QR$LBR            ;MOVE DATA TO LOD BUS REG
000154  012746  001001                          MOV     #<Q$LBD+Q$LBP>,-(SP)    ;CLR DRIVE AND PULSE
000160  052716  000360                          BIS     #<Q$CSEL>,(SP)          ;CLR SELECTION BITS
000164  012746  176000                          MOV     #<Q$NCLK>,-(SP)         ;SET NO-CLOCKS
000170  052716  000260                          BIS     #Q$CP2,(SP)             ;SELECT CP
000174                                          CALL    CSR1
                                        ;
000200  012746  030000                          MOV     #Q$CNC,-(SP)            ;CLEAR CP NO-CLOCK BITS
000204  012746  021000                          MOV     #<Q$CSP+Q$LBD>,-(SP)    ;SET CP CLOCK
000210                                          CALL    CSR1
                                        ;
                                        ;       DE-SELECTION
                                        ;
000214  012746  001001                          MOV     #<Q$LBD+Q$LBP>,-(SP)    ;CLEAR DRIVE AND PULSE
000220  052716  000360                          BIS     #<Q$CSEL>,(SP)          ;CLR SELECTION BITS
000224  012746  176000                          MOV     #<Q$NCLK>,-(SP)         ;SET NO-CLOCKS
000230                                          CALL    CSR1
                                        ;
000234  011666  000002                          MOV     (SP),2(SP)              ;MOVE RETURN ADDRESS DOWN STACK
000240  005726                                  TST     (SP)+                   ;POINT TO RETURN ADDRESS
000242                                          RETURN
```

```
                                  ;
                                  ;       DATA TRANSFER
                                  ;       LOD BUS REGISTER TO A DESTINATION ON THE CP BUS
                                  ;       SINGLE CLOCK SEQUENCER ONLY
                                  ;
                                  ;       INPUT:
                                  ;       2(SP)   DATA FOR PRE-SELECTED CP DESTINATION
                                  ;
000244                            LBCSC::
000244  016667  000002  176424            MOV     2(SP),QR$LBR                ;MOVE DATA TO LOD BUS REG
000252  012746  001001                    MOV     #<Q$LBD+Q$LBP>,-(SP)        ;CLR DRIVE AND PULSE
000256  052716  000360                    BIS     #<Q$CSEL>,(SP)              ;CLR SELECTION BITS
000262  012746  176000                    MOV     #<Q$NCLK>,-(SP)             ;SET NO-CLOCKS
000266  052716  000260                    BIS     #Q$CP2,(SP)                 ;SELECT CP
000272                                    CALL    CSR1
                                  ;
000276  012746  030000                    MOV     #Q$CNC,-(SP)                ;CLEAR CP NO-CLOCK BITS
000302  012746  011000                    MOV     #<Q$CSC+Q$LBD>,-(SP)        ;SET CP CLOCK
000306                                    CALL    CSR1
                                  ;
                                  ;       DE-SELECTION
                                  ;
000312  012746  001001                    MOV     #<Q$LBD+Q$LBP>,-(SP)        ;CLR DRIVE AND PULSE
000316  052716  000360                    BIS     #<Q$CSEL>,(SP)              ;CLR SELECTION BITS
000322  012746  176000                    MOV     #<Q$NCLK>,-(SP)             ;SET NO-CLOCKS
000326                                    CALL    CSR1
                                  ;
000332  011666  000002                    MOV     (SP),2(SP)                  ;MOVE RETURN ADDRESS DOWN STACK
000336  005726                            TST     (SP)+                       ;POINT TO RETURN ADDRESS
000340                                    RETURN
```

```
                                        ;
                                        ;       DATA TRANSFER TO LOD BUS REG FROM CP
                                        ;
                                        ;       OUTPUT:
                                        ;       (SP)    DATA FROM PRE-SELECTED CP SOURCE
                                        ;
                                        CPLB::
000342                                          MOV     #<Q$LBD+Q$LBP>,-(SP)    ;CLR DRIVE AND PULSE
000346  052716  000360                        BIS     #<Q$CSEL>,(SP)          ;CLR SELECTION BITS
000352  012746  176000                        MOV     #<Q$NCLK>,-(SP)         ;SET NO-CLOCKS
000356  052716  000260                        BIS     #Q$CP2,(SP)             ;SOURCE IS CP
000362                                          CALL    CSR1
000366  011646                                  MOV     (SP),-(SP)              ;MOVE RETURN ADDR UP STACK
000370  016766  176424  000002                MOV     QR$LBR,2(SP)            ;MOVE DATA ONTO STACK
                                        ;
000376  012746  000260                        MOV     #Q$CP2,-(SP)            ;CLEAR CP SELECT
000402  005046                                CLR     -(SP)                   ;SET NOTHING
000404                                          CALL    CSR1
000410                                          RETURN
```

```
                                        ;
                                        ;
                                        ;       CP CONTROL·REGISTER·LOADING·
                                        ;
                                        ;       INPUT:
                                        ;       2(SP)   BIT·SETTING·FOR·CP·CONTROL·REGISTER·
                                        ;
000412·                                 CPCR::
000412· 052766  100000  000002·                 BIS·    #CNOBREG,2(SP)          ;AUTOMATIC INHIBIT·OF·BRANCH·REGISTER·
000420                                  CPCRA::
000420  016667  000002  176424                  MOV·    2(SP),QR$LBR·           ;CONTROL·BITS·DESTINED·FOR·CP·
000426  012746  001001                          MOV·    #<Q$LBD+Q$LBP>,-(SP)    ;CLEAR·DRIVE·AND·PULSE·
000432· 052716  000360                          BIS·    #<Q$CSEL>,(SP)          ;CLR·SELECTION·BITS·
000436  012746  000060                          MOV·    #Q$CP,-(SP)             ;SELECT·CP
000442·                                         CALL·   CSR1
                                        ;
000446  005046                                 CLR·    -(SP)                   ;CLEAR·NOTHING·
000450  012746  000001                          MOV·    #Q$LBP,-(SP)            ;SET·PULSE
000454                                          CALL·   CSR1
                                        ;
000460  012746  000061                          MOV·    #<Q$CP+Q$LBP>,-(SP)     ;CLEAR·CR·SELECTION·AND·PULSE·
000464  005046                                 CLR·    -(SP)                   ;SET·NOTHING·
000466                                          CALL·   CSR1
                                        ;
000472· 011666  000002                          MOV·    (SP),2(SP)              ;MOVE·RETURN·ADDRESS DOWN·STACK·;
000476  005726                                 TST·    (SP)+                   ;POINT·TO·RETERN·ADDRESS
000500                                          RETURN·
                                        ;
        000001                          .END·
```

SYMBOL·TABLE·


```
ALUCKE=·040000          BYTE42=·000052·         BYTE94=·000136          PLR.EN=·000200          Q$RNC·=·006000
ALUOE·=·004000          BYTE43=·000053          BYTE95=·000137          QR$CR1=·176420          Q$RSC·=·004000
A01   =·010000          BYTE44=·000054          BYTE96=·000140          QR$CR2=·176422·         Q$RSET=·000010
BITVAL=·000000          BYTE45=·000055          BYTE97=·000141          QR$LBR=·176424          Q$SM  =·100000
BIT0  =·000001          BYTE46=·000056          BYTE98=·000142·         Q$ATTN=·000100          Q$SP  =·000120
BIT1  =·000002·         BYTE47=·000057          BYTE99=·000143          Q$BCL·=·000001          Q$SP2·=·000340
BIT10 =·002000          BYTE48=·000060          BYTVAL=·000144          Q$CCCP=·000040          RGO,EN=·000200
BIT11 =·004000          BYTE49=·000061          CBKALL=·001000          Q$CHB·=·000400          RGO.VA=·020000
BIT12·=·010000          BYTE5 =·000005          CBKCLK=·000400          Q$CHRL=·000200          SEQCS·  000000RG·   002·
BIT13 =·020000          BYTE50=·000062·         CNOBRE=·100000          Q$CLR·=·000040          SEQ.CI=·000010
BIT14 =·040000          BYTE51=·000063          CPCCEN=·010000          Q$CNC·=·030000          S$CLR·=·000000
BIT15 =·100000          BYTE52=·000064          CPCR·   000412RG·   002·Q$CP·  =·000060          S$LA  =·000001
BIT2·  =·000004         BYTE53=·000065          CPCRA·  000420RG·   002·Q$CPCC=·000010          S$OB  =·000005
BIT3  =·000010          BYTE54=·000066          CPLB·   000342RG·   002·Q$CP2·=·000260          S$OR  =·000006
BIT4  =·000020          BYTE55=·000067          CPREAD=·040000          Q$CSC·=·010000          S$OX  =·000004
BIT5  =·000040          BYTE56=·000070          CPSUB·  000000R·    002·Q$CSEL=·000360          S$SR  =·000007
BIT6  =·000100          BYTE57=·000071          CPWRTE=·020000          Q$CSET=·000002·         S$S1  =·000010
BIT7  =·000200          BYTE58=·000072·         CSADRD=·000004          Q$CSP·=·020000          S$S2  =·000014
BIT8·  =·000400         BYTE59=·000073          CSEQCI=·100000          Q$DMA·=·000001          TD$CTR=·176370·
BIT9  =·001000          BYTE6 =·000006          CSOE·  =·000040         Q$ENBK=·040000          TD$CTW=·176360·
BYTE0 =·000000          BYTE60=·000074          CSR1   =·****** GX·     Q$ENOP=·020000          TD$INL=·004000·
BYTE1 =·000001          BYTE61=·000075          CSWRTE=·000100          Q$FAL·=·004000          TD$MEM=·000270
BYTE10=·000012·         BYTE62=·000076          DBR.RD=·000001          Q$FC·  =·000045          TD$OAR=·176344·
BYTE11=·000013          BYTE63=·000077          DB$CPP=·001457          Q$FO·  =·000044          TD$OTR=·176346·
BYTE12=·000014          BYTE64=·000100          DB$SPT=·000026          Q$FP·  =·000046          TD$ORD=·000274
BYTE13=·000015          BYTE65=·000101          DB$TPC=·000023          Q$HBF·=·000002·         TD$SW·=·176376
BYTE14=·000016          BYTE66=·000102·         DISPGS=·100000          Q$ICP·=·000006          TD$TAR=·176372
BYTE15=·000017          BYTE67=·000103          DMAAWR=·000005          Q$IHB·=·000003          TD$TAW=·176362
BYTE16=·000020          BYTE68=·000104          DMARRD=·000003          Q$IHRL=·000002·         TD$TDR=·176374
BYTE17=·000021          BYTE69=·000105          DMARWR=·000004          Q$IMRP=·000007          TD$TDW=·176364
BYTE18=·000022·         BYTE7 =·000007          ENBR·  =·010000         Q$LBD·=·001000          T$AD  =·000020
BYTE19=·000023          BYTE70=·000106          LBCP·   000146RG·   002·Q$LBDP=·001001          T$BA  =·000002
BYTE2·=·000002·         BYTE71=·000107          LBCSC·  000244RG·   002·Q$LBP·=·000001          T$BD  =·000010
BYTE20=·000024          BYTE72=·000110          LOC.EN=·000100          Q$LDCD=·000003          T$BSO·=·100000
BYTE21=·000025          BYTE73=·000111          LOC.WA=·040000          Q$LDMD=·000004          T$BT  =·000020
BYTE22=·000026          BYTE74=·000112·         LOC.WB=·100000          Q$LDPP=·002000          T$BTAR=·000030
BYTE23=·000027          BYTE75=·000113          MAREN1=·000001          Q$LHP·=·010000          T$BTD·=·002000
BYTE24=·000030          BYTE76=·000114          MAREN2=·004000          Q$MNC·=·140000          T$CD  =·000100
BYTE25=·000031          BYTE77=·000115          MARLOD=·010000          Q$MR·  =·000052·         T$CLK·=·002000
BYTE26=·000032·         BYTE78=·000116          MAROUT=·000002·         Q$MRP·=·000040          T$DISK=·000200
BYTE27=·000033          BYTE79=·000117          MAR.LO=·002000          Q$MRP2=·000240          T$DRD·=·000004
BYTE28=·000034          BYTE8 =·000010          MAR.OU=·000040          Q$MSC·=·040000          T$EMEM=·010000
BYTE29=·000035          BYTE80=·000120          MBKALL=·001000          Q$MSET=·000004          T$FSAA=·000000
BYTE3 =·000003          BYTE81=·000121          MBKCLK=·000400          Q$MSP·=·100000          T$FSAB=·000004
BYTE30=·000036          BYTE82=·000122·         MMADRD=·000100          Q$NCLK=·176000          T$FSAC=·000014
BYTE31=·000037          BYTE83=·000123          MMLEFT=·000002·         Q$PP·  =·000100          T$FSB2=·000010·
BYTE32=·000040          BYTE84=·000124          MMOE·  =·000004         Q$PPSW=·000320          T$IB  =·000026
BYTE33=·000041          BYTE85=·000125          MMWRTE=·000010          Q$PP2·=·000300          T$IBAR=·000024
BYTE34=·000042·         BYTE86=·000126          MNOBRE=·100000          Q$QHLT=·000013          T$IBE·=·020000
BYTE35=·000043          BYTE87=·000127          MREN1 =·000001          Q$QL·  =·000043          T$IBF·=·040000
BYTE36=·000044          BYTE88=·000130          MREN2·=·020000          Q$QLA·=·000053          T$ICD·=·000040
BYTE37=·000045          BYTE89=·000131          MSYN·  =·000040         Q$QLB·=·000054          T$MODE=·004000
BYTE38=·000046          BYTE9 =·000011          N·     =·000144         Q$QLR·=·000001          T$OB  =·000036·
BYTE39=·000047          BYTE90=·000132·         PLB·   =·000010         Q$QW·  =·000042·         T$OBE·=·004000
BYTE4 =·000004          BYTE91=·000133          PLC·   =·000020         Q$RDCD=·000005          T$OBF·=·010000
BYTE40=·000050          BYTE92=·000134          PLD·   =·000030         Q$RDMD=·000006          T$OBRA=·000034
BYTE41=·000051          BYTE93=·000135          PLRWR·=·000200          Q$REBK=·001000          T$OBWA=·000032
```

```
CPSUB  MACRO M1110  27-MAR-80 14:45  PAGE 10-2
SYMBOL TABLE
```

```
T$OUTA= 100000          WORD18= 000044          WORD4 = 000010          WORD61= 000172          WORD82= 000244
T$RBDO= 000200          WORD19= 000046          WORD40= 000120          WORD62= 000174          WORD83= 000246
T$RNB = 000040          WORD2 = 000004          WORD41= 000122          WORD63= 000176          WORD84= 000250
T$RSET= 040000          WORD20= 000050          WORD42= 000124          WORD64= 000200          WORD85= 000252
T$SC  = 000022          WORD21= 000052          WORD43= 000126          WORD65= 000202          WORD86= 000254
T$SCLK= 020000          WORD22= 000054          WORD44= 000130          WORD66= 000204          WORD87= 000256
T$SEG1= 000000          WORD23= 000056          WORD45= 000132          WORD67= 000206          WORD88= 000260
T$SEG2= 000001          WORD24= 000060          WORD46= 000134          WORD68= 000210          WORD89= 000262
T$SEG3= 000002          WORD25= 000062          WORD47= 000136          WORD69= 000212          WORD9 = 000022
T$SO  = 000001          WORD26= 000064          WORD48= 000140          WORD7 = 000016          WORD90= 000264
T$UBUS= 100000          WORD27= 000066          WORD49= 000142          WORD70= 000214          WORD91= 000266
T$1CLK= 000400          WORD28= 000070          WORD5 = 000012          WORD71= 000216          WORD92= 000270
T$8BEN= 000020          WORD29= 000072          WORD50= 000144          WORD72= 000220          WORD93= 000272
UBD.IN= 000020          WORD3 = 000006          WORD51= 000146          WORD73= 000222          WORD94= 000274
WORD0 = 000000          WORD30= 000074          WORD52= 000150          WORD74= 000224          WORD95= 000276
WORD1 = 000002          WORD31= 000076          WORD53= 000152          WORD75= 000226          WORD96= 000300
WORD10= 000024          WORD32= 000100          WORD54= 000154          WORD76= 000230          WORD97= 000302
WORD11= 000026          WORD33= 000102          WORD55= 000156          WORD77= 000232          WORD98= 000304
WORD12= 000030          WORD34= 000104          WORD56= 000160          WORD78= 000234          WORD99= 000306
WORD13= 000032          WORD35= 000106          WORD57= 000162          WORD79= 000236          WRDVAL= 000310
WORD14= 000034          WORD36= 000110          WORD58= 000164          WORD8 = 000020          WRTCS   000066RG    002
WORD15= 000036          WORD37= 000112          WORD59= 000166          WORD80= 000240          XTREAD= 001000
WORD16= 000040          WORD38= 000114          WORD6 = 000014          WORD81= 000242          XTWRTE= 000400
WORD17= 000042          WORD39= 000116          WORD60= 000170

. ABS.  000000     000
        000000     001
CPSUB   000502     002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3021 WORDS  ( 12 PAGES)
DYNAMIC MEMORY:  3860 WORDS  ( 14 PAGES)
ELAPSED TIME:  00:00:43
CPSUB,@PSUB/-SP=[20,1]IM,[20,1]CPSUB
```

```
                                        .TITLE MRPSUB
000000                                  .PSECT MRPSUB

                                ;
                                ;       HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
                                ;       *******   PROTOTYPE VERSION  *******
                                ;
                                ;       MRP SUBROUTINES
                                ;
                                ;       CONTROL STORE SEQUENCING (MATCH REPORT PROCESSOR)
                                ;       JUMP TO START ADDRESS (VIA BRANCH REGISTER)
                                ;
                                ;       INPUT:
                                ;       2(SP)   START ADDRESS
                                ;
                                ;       OUTPUT:
                                ;       MRP SEQUENCER SET TO START ADDRESS
                                ;
000000                          MRPSUB:
000000                          SEQMM::
                                ;
                                ;       STOP CP AND MRP CLOCKS
                                ;
000000  005046                          CLR     -(SP)                   ;CLEAR NOTHING
000002  012746  176000                  MOV     #Q$NCLK,-(SP)           ;SET NO CLOCKS
000006                                  CALL    CSR1
                                ;
                                ;       GO TO LOAD MODE (AND CLEAR EVERYTHING ELSE)
                                ;
000012  005067  176422                  CLR     QR$CR2                  ;DO IT
                                ;
                                ;       SELECT BRANCH REGISTER IN MRP
                                ;
000016  005046                          CLR     -(SP)                   ;DEFINE BR REG SELECT
000020                                  CALL    MRPCRA                  ;WRITE CR (WITHOUT AUTOMATIC BR SUPPRESS)
                                ;
                                ;       CLOCK MRP ONCE
                                ;
000024  012746  140000                  MOV     #Q$MNC,-(SP)            ;CLEAR MRP NO CLOCKS
000030  012746  100000                  MOV     #Q$MSP,-(SP)            ;SET MRP SINGLE CLOCK
000034                                  CALL    CSR1
                                ;
                                ;       MOVE INSTRUCTION AND ADDRESS TO BRANCH REGISTER VIA LOAD BUS
                                ;
000040  052766  020000  000002          BIS     #BIT13,2(SP)            ;OR JMP INSRUCTION (02) INTO HIGH ORDER NIBBLE
000046  016646  000002                  MOV     2(SP),-(SP)             ;COPY ARGUMENT INTO STACK FOR LBMRP CALL
000052                                  CALL    LBMRP                   ;DO MOVE - FIRST CLOCK
                                ;
000056  011666  000002                  MOV     (SP),2(SP)              ;MOVE RETURN ADDRESS DOWN STACK
000062  005726                          TST     (SP)+                   ;POINT TO RETURN ADDRESS
000064                                  RETURN
```

```
                                ;
                                ;       CONTROL·STORE·SEQUENCING·(MATCH·REPORT·PROCESSOR)
                                ;       SET·START·ADDRESS·FOR·WRITE·
                                ;
                                ;       INPUT:
                                ;       2(SP)   START·ADDRESS·
                                ;
                                ;       OUTPUT·
                                ;       MRP·SEQUENCER·SET·TO·START·ADDRESS·
                                ;
000066                          WRTMM::
                                ;
                                ;       STOP·CP·AND·MRP·CLOCKS
                                ;
000066  005046                          CLR·    -(SP)                   ;CLEAR·NOTHING·
000070  012746  176000                  MOV·    #Q$NCLK,-(SP)           ;SET·NO·CLOCKS·
000074                                  CALL·   CSR1
                                ;
                                ;       GO TO·LOAD·MODE·(AND·CLEAR·EVERYTHING·ELSE)
                                ;
000100  005067  176422                  CLR·    QR$CR2·                 ;DO·IT
                                ;
                                ;       SELECT·BRANCH·REGISTER IN·MRP·
                                ;
000104  005046                          CLR·    -(SP)                   ;DEFINE·BR REG·SELECT·
000106                                  CALL·   MRPCRA·                 ;WRITE·CR·(WITHOUT AUTOMATIC·BR·INHIBIT)
                                ;
                                ;       CLOCK·MRP·ONCE·
                                ;
000112· 012746  140000·                 MOV·    #Q$MNC,-(SP)            ;CLEAR·MRP NO·CLOCKS
000116  012746  100000·                 MOV·    #Q$MSP,-(SP)            ;SET·MRP·SINGLE·CLOCK·
000122·                                 CALL·   CSR1
                                ;
                                ;       MOVE·INSTRUCTION·AND·ADDRESS·TO·BRANCH·REGISTER·VIA·LOAD·BUS·
                                ;
000126  052766  020000· 000002·         BIS·    #BIT13,2(SP)            ;OR·JMP·INSRUCTION·(02) INTO·HIGH·ORDER·NIBBLE·
000134  016646  000002·                 MOV·    2(SP),-(SP)             ;COPY·ARGUMENT·INTO·STACK·FOR·LBMRP·CALL
000140                                  CALL·   LBMRP·                  ;DO·MOVE·- FIRST·CLOCK·
                                ;
000144  011666  000002·                 MOV·    (SP),2(SP)              ;MOVE·RETURN·ADDRESS DOWN·STACK·
000150  005726                          TST·    (SP)+                   ;POINT·TO·RETURN·ADDRESS·
```

```
 94                                     ;
 95                                     ;       DATA·TRANSFER·
 96                                     ;       LOD·BUS·REGISTER·TO·A·DESTINATION·ON·THE·MRP·BUS·
 97                                     ;
 98                                     ;       INPUT:
 99                                     ;       2(SP)    DATA FOR·PRE-SELECTED·MRP·DESTINATION·
100                                     ;
101 000152·                            LBMRP::
102·000152· 016667  000002  176424             MOV·     2(SP),QR$LBR·              ;MOVE·DATA TO·LOD·BUS·REG·
103 000160  012746  001001                     MOV·     #<Q$LBD+Q$LBP>,-(SP)       ;CLR·DRIVE AND·PULSE
104 000164  052716  000360·                    BIS·     #<Q$CSEL>,(SP)             ;CLR·SELECTION·BITS·
105 000170  012746  176000·                    MOV·     #<Q$NCLK>,-(SP)            ;SET·NO-CLOCKS·
106 000174  052716  000240                     BIS·     #Q$MRP2,(SP)               ;SELECT·MRP·
107 000200                                     CALL·    CSR1
108                                     ;
109 000204  012746  140000                     MOV·     #Q$MNC,-(SP)               ;CLEAR·MRP NO-CLOCK·BITS·
110 000210  012746  101000·                    MOV·     #<Q$MSP+Q$LBD>,-(SP)       ;SET·MRP·CLOCK·
111 000214                                     CALL·    CSR1
112·                                    ;
113                                     ;       DE-SELECTION
114                                     ;
115 000220  012746  001001                     MOV·     #<Q$LBD+Q$LBP>,-(SP)       ;CLEAR·DRIVE·AND·PULSE·
116 000224  052716  000360                     BIS·     #<Q$CSEL>,(SP)             ;CLR·SELECTION·BITS·
117 000230  012746  176000                     MOV·     #<Q$NCLK>,-(SP)            ;SET·NO-CLOCKS·
118 000234                                     CALL·    CSR1
119                                     ;
120 000240  011666  000002                     MOV·     (SP),2(SP)                 ;MOVE·RETURN·ADDRESS DOWN·STACK·
121 000244  005726                             TST·     (SP)+                      ;POINT·TO·RETURN·ADDRESS
122·000246                                     RETURN·
```

```
                                        ;
                                        ;       DATA TRANSFER
                                        ;       LOD BUS REGISTER TO A DESTINATION ON THE MRP BUS
                                        ;       SINGLE CLOCK SEQUENCER ONLY
                                        ;
                                        ;       INPUT:
                                        ;       2(SP)   DATA FOR PRE-SELECTED MRP DESTINATION
                                        ;
000250                                  LBMSC::
000250  016667  000002  176424          MOV     2(SP),QR$LBR            ;MOVE DATA TO LOD BUS REG
000256  012746  001001                  MOV     #<Q$LBD+Q$LBP>,-(SP)    ;CLR DRIVE AND PULSE
000262  052716  000360                  BIS     #<Q$CSEL>,(SP)          ;CLR SELECTION BITS
000266  012746  176000                  MOV     #<Q$NCLK>,-(SP)         ;SET NO-CLOCKS
000272  052716  000240                  BIS     #Q$MRP2,(SP)            ;SELECT MRP
000276                                  CALL    CSR1
                                        ;
000302  012746  140000                  MOV     #Q$MNC,-(SP)            ;CLEAR MRP NO-CLOCK BITS
000306  012746  041000                  MOV     #<Q$MSC+Q$LBD>,-(SP)    ;SET MRP CLOCK
000312                                  CALL    CSR1
                                        ;
                                        ;       DE-SELECTION
                                        ;
000316  012746  001001                  MOV     #<Q$LBD+Q$LBP>,-(SP)    ;CLR DRIVE AND PULSE
000322  052716  000360                  BIS     #<Q$CSEL>,(SP)          ;CLR SELECTION BITS
000326  012746  176000                  MOV     #<Q$NCLK>,-(SP)         ;SET NO-CLOCKS
000332                                  CALL    CSR1
                                        ;
000336  011666  000002                  MOV     (SP),2(SP)              ;MOVE RETURN ADDRESS DOWN STACK
000342  005726                          TST     (SP)+                   ;POINT TO RETURN ADDRESS
000344                                  RETURN
```

```
                                        ;
                                        ;       DATA·TRANSFER·TO·LOD·BUS·REG·FROM·MRP·
                                        ;
                                        ;       OUTPUT:
                                        ;       (SP)    DATA FROM·PRE-SELECTED·MRP·SOURCE·
                                        ;
000346                                  MRPLB::
000346  012746  001001                          MOV·    #<Q$LBD+Q$LBP>,-(SP)    ;CLR·DRIVE AND·PULSE
000352  052716  000360                          BIS·    #<Q$CSEL>,(SP)          ;CLR·SELECTION·BITS·
000356  012746  176000                          MOV·    #<Q$NCLK>,-(SP)         ;SET·NO-CLOCKS·
000362  052716  000240                          BIS·    #Q$MRP2,(SP)            ;SOURCE·IS MRP·
000366                                          CALL·   CSR1
                                        ;
000372  011646                                  MOV·    (SP),-(SP)              ;MOVE·RETURN·ADDR·UP STACK·
000374  016766  176424  000002                  MOV·    QR$LBR,2(SP)            ;MOVE·DATA ONTO·STACK·
                                        ;
000402  012746  000240                          MOV·    #Q$MRP2,-(SP)           ;CLEAR·MRP SELECT·
000406  005046                                  CLR·    -(SP)                   ;SET·NOTHING·
000410                                          CALL·   CSR1
000414                                          RETURN·
```

```
                                ;
                                ;
                                ;       MRP·CONTROL·REGISTER·LOADING·
                                ;
                                ;       INPUT:
                                ;       2(SP)   BIT·SETTING·FOR·MRP·CONTROL·REGISTER·
                                ;
000416                          MRPCR::
000416  052766  100000  000002·         BIS·    #MNOBREG,2(SP)          ;INHIBIT·BRANCH·REGISTER·SELECT·
000424                          MRPCRA::
000424  016667  000002  176424           MOV·    2(SP),@R$LBR·           ;CONTROL·BITS·DESTINED·FOR·MRP
000432· 012746  001001                   MOV·    #<Q$LBD+Q$LBP>,-(SP)    ;CLEAR·DRIVE·AND·PULSE·
000436  052716  000360                   BIS·    #<Q$CSEL>,(SP)          ;CLR·SELECTION·BITS·
000442· 012746  000040·                  MOV·    #Q$MRP,-(SP)            ;SELECT·MRP·
000446                                   CALL·   CSR1
                                ;
000452· 005046                           CLR·    -(SP)                   ;CLEAR·NOTHING·
000454  012746  000001                   MOV·    #Q$LBP,-(SP)            ;SET·PULSE
000460                                   CALL·   CSR1
                                ;
000464  012746  000041                   MOV·    #<Q$MRP+Q$LBP>,-(SP)    ;CLEAR·CR·SELECTION·AND·PULSE·
000470  005046                           CLR·    -(SP)                   ;SET·NOTHING·
000472·                                  CALL·   CSR1
                                ;
000476  011666  000002                   MOV·    (SP),2(SP)              ;MOVE·RETURN·ADDRESS·DOWN·STACK·
000502· 005726                           TST·    (SP)+                   ;POINT·TO·RETURN·ADDRESS
000504                                   RETURN·
                                ;
        000001                           .END·
```

```
ALUCKE=·040006          BYTE42=·000052·         BYTE94=·000136          PLR.EN=·000200          Q$RNC·=·006000
ALUOE·=·004000          BYTE43=·000053          BYTE95=·000137          QR$CR1=·176420          Q$RSC·=·004000
A01    =·010000         BYTE44=·000054          BYTE96=·000140          QR$CR2=·176422·         Q$RSET=·000010
BITVAL=·000000          BYTE45=·000055          BYTE97=·000141          QR$LBR=·176424          Q$SM  =·100000
BIT0   =·000001         BYTE46=·000056          BYTE98=·000142·         Q$ATTN=·000100          Q$SP  =·000120
BIT1   =·000002·        BYTE47=·000057          BYTE99=·000143          Q$BCL·=·000001          Q$SP2·=·000340
BIT10  =·002000         BYTE48=·000060          BYTVAL=·000144          Q$CCCP=·000040          RGO.EN=·000200
BIT11  =·004000         BYTE49=·000061          CBKALL=·001000          Q$CHB·=·000400          RGO.VA=·020000
BIT12·=·010000          BYTE5  =·000005         CBKCLK=·000400          Q$CHRL=·000200          SEQMM1·=·000000RG·    002·
BIT13  =·020000         BYTE50=·000062·         CNOBRE=·100000          Q$CLR·=·000040          SEQ.CI=·000010
BIT14  =·040000         BYTE51=·000063          CPCCEN=·010000          Q$CNC·=·030000          S$CLR·=·000000
BIT15  =·100000         BYTE52=·000064          CPREAD=·040000          Q$CP·=·000060           S$LA  =·000001
BIT2·  =·000004         BYTE53=·000065          CPWRTE=·020000          Q$CPCC=·000010          S$QB  =·000005
BIT3   =·000010         BYTE54=·000066          CSADRD=·000004          Q$CP2·=·000260          S$QR  =·000006
BIT4   =·000020         BYTE55=·000067          CSEQCI=·100000          Q$CSC·=·010000          S$QX  =·000004
BIT5   =·000040         BYTE56=·000070          CSOE·=·000040           Q$CSEL=·000360          S$SR  =·000007
BIT6   =·000100         BYTE57=·000071          CSR1   =·****** GX·     Q$CSET=·000002·         S$S1  =·000010
BIT7   =·000200         BYTE58=·000072·         CSWRTE=·000100          Q$CSP·=·020000          S$S2  =·000014
BIT8·  =·000400         BYTE59=·000073          DBR.RD=·000001          Q$DMA·=·000001          TD$CTR=·176370
BIT9   =·001000         BYTE6  =·000006         DB$CPP=·001457          Q$ENBK=·040000          TD$CTW=·176360
BYTE0  =·000000         BYTE60=·000074          DB$SPT=·000026          Q$ENOP=·020000          TD$INL=·004000
BYTE1  =·000001         BYTE61=·000075          DB$TPC=·000023          Q$FAL·=·004000          TD$MEM=·000270
BYTE10=·000012·         BYTE62=·000076          DISPGS=·100000          Q$FC·=·000045           TD$OAR=·176344
BYTE11=·000013          BYTE63=·000077          DMAAWR=·000005          Q$FO·=·000044           TD$OTR=·176346
BYTE12=·000014          BYTE64=·000100          DMARRD=·000003          Q$FP·=·000046           TD$QRD=·000274
BYTE13=·000015          BYTE65=·000101          DMARWR=·000004          Q$HBF·=·000002·         TD$SW·=·176376
BYTE14=·000016          BYTE66=·000102·         ENBR·=·010000           Q$ICP·=·000006          TD$TAR=·176372
BYTE15=·000017          BYTE67=·000103          LBMRP·  000152RG·  002· Q$IHB·=·000003          TD$TAW=·176362
BYTE16=·000020          BYTE68=·000104          LBMSC·  000250RG·  002· Q$IHRL=·000002·         TD$TDR=·176374
BYTE17=·000021          BYTE69=·000105          LOC.EN=·000100          Q$IMRP=·000007          TD$TDW=·176364
BYTE18=·000022·         BYTE7  =·000007         LOC.WA=·040000          Q$LBD·=·001000          T$AD  =·000020
BYTE19=·000023          BYTE70=·000106          LOC.WB=·100000          Q$LBDP=·001001          T$BA  =·000002
BYTE2·=·000002·         BYTE71=·000107          MAREN1=·000001          Q$LBP·=·000001          T$BD  =·000010
BYTE20=·000024          BYTE72=·000110          MAREN2=·004000          Q$LDCD=·000003          T$BSO·=·100000
BYTE21=·000025          BYTE73=·000111          MARLOD=·010000          Q$LDMD=·000004          T$BT  =·000020
BYTE22=·000026          BYTE74=·000112·         MAROUT=·000002·         Q$LDPP=·002000          T$BTAR=·000030
BYTE23=·000027          BYTE75=·000113          MAR.LO=·002000          Q$LHP·=·010000          T$BTD·=·002000
BYTE24=·000030          BYTE76=·000114          MAR.OU=·000040          Q$MNC·=·140000          T$CD  =·000100
BYTE25=·000031          BYTE77=·000115          MBKALL=·001000          Q$MR·=·000052·          T$CLK·=·002000
BYTE26=·000032·         BYTE78=·000116          MBKCLK=·000400          Q$MRP·=·000040          T$DISK=·000200
BYTE27=·000033          BYTE79=·000117          MMADRD=·000100          Q$MRP2=·000240          T$DRD·=·000004
BYTE28=·000034          BYTE8  =·000010         MMLEFT=·000002·         Q$MSC·=·040000          T$EMEM=·010000
BYTE29=·000035          BYTE80=·000120          MMOE·=·000004           Q$MSET=·000004          T$FSAA=·000000
BYTE3  =·000003         BYTE81=·000121          MMWRTE=·000010          Q$MSR·=·100000          T$FSAB=·000004
BYTE30=·000036          BYTE82=·000122·         MNOBRE=·100000          Q$NCLK=·176000          T$FSAC=·000014
BYTE31=·000037          BYTE83=·000123          MREN1  =·000001         Q$PP·=·000100           T$FSB2=·000010
BYTE32=·000040          BYTE84=·000124          MREN2·=·020000          Q$PPSW=·000320          T$IB  =·000026
BYTE33=·000041          BYTE85=·000125          MRPCR·  000416RG·  002· Q$PP2·=·000300          T$IBAR=·000024
BYTE34=·000042·         BYTE86=·000126          MRPCRA  000424RG·  002· Q$QHLT=·000013          T$IBE·=·020000
BYTE35=·000043          BYTE87=·000127          MRPLB·  000346RG·  002· Q$QL·=·000043           T$IBF·=·040000
BYTE36=·000044          BYTE88=·000130          MRPSUB  000000R·   002· Q$QLA·=·000053          T$ICD·=·000040
BYTE37=·000045          BYTE89=·000131          MSYN·=·000040           Q$QLB·=·000054          T$MODE=·004000
BYTE38=·000046          BYTE9  =·000011         N·     =·000144         Q$QLR·=·000001          T$OB  =·000036
BYTE39=·000047          BYTE90=·000132·         PLB·   =·000010         Q$QW·=·000042·          T$OBE·=·004000
BYTE4  =·000004         BYTE91=·000133          PLC·   =·000020         Q$RDCD=·000005          T$OBF·=·010000
BYTE40=·000050          BYTE92=·000134          PLD·   =·000030         Q$RDMD=·000006          T$OBRA=·000034
BYTE41=·000051          BYTE93=·000135          PLRWR·=·000200          Q$REBK=·001000          T$OBWA=·000032
```

SYMBOL TABLE

```
T$0UTA= 100000          WORD18= 000044          WORD4 = 000010          WORD61= 000172          WORD82= 000244
T$RBDO= 000200          WORD19= 000046          WORD40= 000120          WORD62= 000174          WORD83= 000246
T$RNB = 000040          WORD2 = 000004          WORD41= 000122          WORD63= 000176          WORD84= 000250
T$RSET= 040000          WORD20= 000050          WORD42= 000124          WORD64= 000200          WORD85= 000252
T$SC  = 000022          WORD21= 000052          WORD43= 000126          WORD65= 000202          WORD86= 000254
T$SCLK= 020000          WORD22= 000054          WORD44= 000130          WORD66= 000204          WORD87= 000256
T$SEG1= 000000          WORD23= 000056          WORD45= 000132          WORD67= 000206          WORD88= 000260
T$SEG2= 000001          WORD24= 000060          WORD46= 000134          WORD68= 000210          WORD89= 000262
T$SEG3= 000002          WORD25= 000062          WORD47= 000136          WORD69= 000212          WORD9 = 000022
T$SO  = 000001          WORD26= 000064          WORD48= 000140          WORD7 = 000016          WORD90= 000264
T$UBUS= 100000          WORD27= 000066          WORD49= 000142          WORD70= 000214          WORD91= 000266
T$1CLK= 000400          WORD28= 000070          WORD5 = 000012          WORD71= 000216          WORD92= 000270
T$8BEN= 000020          WORD29= 000072          WORD50= 000144          WORD72= 000220          WORD93= 000272
UBD.IN= 000020          WORD3 = 000006          WORD51= 000146          WORD73= 000222          WORD94= 000274
WORD0 = 000000          WORD30= 000074          WORD52= 000150          WORD74= 000224          WORD95= 000276
WORD1 = 000002          WORD31= 000076          WORD53= 000152          WORD75= 000226          WORD96= 000300
WORD10= 000024          WORD32= 000100          WORD54= 000154          WORD76= 000230          WORD97= 000302
WORD11= 000026          WORD33= 000102          WORD55= 000156          WORD77= 000232          WORD98= 000304
WORD12= 000030          WORD34= 000104          WORD56= 000160          WORD78= 000234          WORD99= 000306
WORD13= 000032          WORD35= 000106          WORD57= 000162          WORD79= 000236          WRDVAL= 000310
WORD14= 000034          WORD36= 000110          WORD58= 000164          WORD8 = 000020          WRTMM   000066RG    002
WORD15= 000036          WORD37= 000112          WORD59= 000166          WORD80= 000240          XTREAD= 001000
WORD16= 000040          WORD38= 000114          WORD6 = 000014          WORD81= 000242          XTWRTE= 000400
WORD17= 000042          WORD39= 000116          WORD60= 000170

. ABS.  000000     000
        000000     001
MRPSUB  000506     002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3021 WORDS  ( 12 PAGES)
DYNAMIC MEMORY:  3860 WORDS  ( 14 PAGES)
ELAPSED TIME:  00:00:43
MRPSUB,MRPSUB/-SP=[20,1]IM,[20,1]MRPSUB
```

```
                                        .TITLE  PPSUB
000000                                  .PSECT  PPSUB
                                ;
                                        .MCALL  WTSE$S,CLEF$S
                                ;
          000003                EFN.3   =       3
                                ;
                                ;
                                ;       HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
                                ;       PIPELINE PROCESSORS SUBROUTINES
                                ;
                                ;
                                ;       DATA TRANSFER
                                ;       LOD BUS REGISTER TO A DESTINATION ON THE PPS BUS
                                ;
                                ;       INPUT:
                                ;       2(SP)   DATA FOR PRE-SELECTED PPS DESTINATION
                                ;
                                ;
000000                          LBPP::
000000  016667  000002  176424          MOV     2(SP),@R$LBR            ;MOVE DATA TO LOD BUS REG
000006  012746  001001                MOV     #<@Q$LBD+@Q$LBP>,-(SP)  ;CLR DRIVE AND PULSE
000012  052716  000360                BIS     #<@Q$CSEL>,(SP)         ;CLR SELECTION BITS
000016  012746  176000                MOV     #<@Q$NCLK>,-(SP)        ;SET NO-CLOCKS
000022  052716  000300                BIS     #@Q$PP2,(SP)            ;SELECT PP
000026                                CALL    CSR1                    ;
                                ;
000032  012746  006000                MOV     #@Q$RNC,-(SP)           ;CLEAR CP NO-CLOCK BITS
000036  012746  001000                MOV     #@Q$LBD,-(SP)           ;SET LOD BUS DRIVE
000042                                CALL    CSR1                    ;
                                ;
                                ;       DE-SELECTION
                                ;
000046  012746  001001                MOV     #<@Q$LBD+@Q$LBP>,-(SP)  ;CLEAR DRIVE AND PULSE
000052  052716  000360                BIS     #<@Q$CSEL>,(SP)         ;CLR SELECTION BITS
000056  012746  176000                MOV     #<@Q$NCLK>,-(SP)        ;SET NO-CLOCKS
000062                                CALL    CSR1                    ;
                                ;
000066  011666  000002                MOV     (SP),2(SP)              ;MOVE RETURN ADDRESS DOWN STACK
000072  005726                        TST     (SP)+                   ;POINT TO RETURN ADDRESS
000074                                RETURN
```

```
                                ;
                                ;
                                ;       DATA TRANSFER
                                ;       LOD BUS REGISTER TO A DESTINATION ON THE PPS BUS
                                ;       SINGLE CLOCK SEQUENCER ONLY
                                ;
                                ;       INPUT:
                                ;       2(SP)   DATA FOR PRE-SELECTED PPS DESTINATION
                                ;
                                ;
000076                          LBPSC::
000076  016667  000002  176424          MOV     2(SP),QR$LBR            ;MOVE DATA TO LOD BUS REG
000104  012746  001001                  MOV     #<Q$LBD+Q$LBP>,-(SP)    ;CLEAR DRIVE AND PULSE
000110  052716  000360                  BIS     #Q$CSEL,(SP)            ;CLR SELECTION BITS
000114  012746  176000                  MOV     #<Q$NCLK>,-(SP)         ;SET NO-CLOCKS
000120  052716  000300                  BIS     #Q$PP2,(SP)             ;SELECT PP
000124                                  CALL    CSR1                    ;WRITE CONTROL REGISTER
                                ;
000130  012746  006000                  MOV     #Q$RNC,-(SP)            ;CLEAR PP NO-CLOCK BITS
000134  012746  005000                  MOV     #<Q$RSC+Q$LBD>,-(SP)    ;SET PP CLOCK
000140                                  CALL    CSR1                    ;
                                ;
                                ;       DE-SELECTION
                                ;
000144  012746  001001                  MOV     #<Q$LBD+Q$LBP>,-(SP)    ;CLEAR DRIVE AND PULSE
000150  052716  000360                  BIS     #Q$CSEL,(SP)            ;CLR SELECTION BITS
000154  012746  176000                  MOV     #<Q$NCLK>,-(SP)         ;SET NO CLOCKS
000160                                  CALL    CSR1
                                ;
000164  011666  000002                  MOV     (SP),2(SP)              ;MOVE RETURN ADDRESS DOWN STACK
000170  005726                          TST     (SP)+                   ;POINT TO RETURN ADDRESS
000172                                  RETURN
```

```
                                        ;
                                        ;
                                        ;       DATA TRANSFER TO LOD BUS REG FROM PPS.
                                        ;
                                        ;       OUTPUT:
                                        ;       (SP)    DATA FROM PRE-SELECTED PPS SOURCE.
                                        ;
                                        ;
000174                                  PPLB::
000174  012746  001001                          MOV     #<Q$LBD+Q$LBP>,-(SP)    ;CLEAR DRIVE AND PULSE
000200  052716  000360                          BIS     #Q$CSEL,(SP)            ;CLR SELECTION BITS
000204  012746  176000                          MOV     #<Q$NCLK>,-(SP)         ;SET NO-CLOCKS
000210  052716  000300                          BIS     #Q$PP2,(SP)             ;SOURCE IS PP
000214                                          CALL    CSR1
000220  011646                                  MOV     (SP),-(SP)              ;MOVE RETURN ADDR UP STACK
000222  016766  176424  000002                  MOV     QR$LBR,2(SP)            ;MOVE DATA ONTO STACK
                                        ;
000230  012746  000300                          MOV     #Q$PP2,-(SP)            ;CLEAR PP SELECT
000234  005046                                  CLR     -(SP)                   ;SET NOTHING
000236                                          CALL    CSR1                    ;
000242                                  RETURN
```

```
                                        ;
                                        ;
                                        ;       PPS CONTROL REGISTER LOADING
                                        ;
                                        ;       INPUT:
                                        ;       2(SP)   BIT SETTING FOR PPS CONTROL REGISTER
                                        ;
                                        ;
000244                                  PPCR::
000244  016667  000002  176424                  MOV     2(SP),QR$LBR            ;CONTROL BITS DESTINED FOR PPS
000252  012746  001001                          MOV     #<Q$LBD+Q$LBP>,-(SP)    ;CLEAR DRIVE AND PULSE
000256  052716  000360                          BIS     #Q$CSEL,(SP)            ;CLR SELECTION BITS
000262  012746  000100                          MOV     #Q$PP,-(SP)             ;SELECT PP
000266                                          CALL    CSR1                    ;
                                        ;
000272  005046                                  CLR     -(SP)                   ;CLEAR NOTHING
000274  012746  000001                          MOV     #Q$LBP,-(SP)            ;SET PULSE
000300                                          CALL    CSR1                    ;
                                        ;
000304  012746  000101                          MOV     #<Q$PP+Q$LBP>,-(SP)     ;CLEAR CR SELECTION AND PULSE
000310  005046                                  CLR     -(SP)                   ;SET NOTHING
000312                                          CALL    CSR1
                                        ;
000316  011666  000002                          MOV     (SP),2(SP)              ;MOVE RETURN ADDRESS DOWN STACK
000322  005726                                  TST     (SP)+                   ;POINT TO RETURN ADDRESS
000324                                          RETURN
```

```
                                        ;
                                        ;
                                        ;       SELECT A QLB PAGE
                                        ;
                                        ;       UPON ENTRY:
                                        ;       2(SP) = PAGE NUMBER
                                        ;       UPON EXIT:
                                        ;       STACK RESTORED
                                        ;       R1, R2 DESTROYED
                                        ;
                                        ;
000326                                  SELPG::
000326  005046                                  CLR     -(SP)                   ;START MICROCODE AT 0
000330                                          CALL    SEQCS
000334  005046                                  CLR     -(SP)                   ;RESET BR INHIBIT
000336                                          CALL    CPCR
000342  012746  000377                          MOV     #377,-(SP)              ;SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
000346                                          CALL    SEQMM
000352  005046                                  CLR     -(SP)                   ;RESET BR INHIBIT
000354                                          CALL    MRPCR
000360  012767  001000  176422                  MOV     #Q$REBK,QR$CR2          ;RE-ARM INTERRUPTS
000366  012767  120000  176422                  MOV     #<Q$SM+Q$ENOP>,QR$CR2   ;SET SEARCH MODE + ENABLE INTERRUPTS
000374  012746  000360                          MOV     #Q$CSEL,-(SP)           ;CLEAR ALL SELECTIONS
000400  052716  001001                          BIS     #<Q$LBD+Q$LBP>,(SP)     ;CLEAR DRIVE AND PULSE
000404  052716  030000                          BIS     #Q$CNC,(SP)             ;CLEAR CP NO-CLOCK
000410  005046                                  CLR     -(SP)                   ;SET NOTHING
000412                                          CALL    CSR1
                                        ;
000416  012767  000003  176424                  MOV     #Q$LDCD,QR$LBR          ;MOVE ATTN CODE TO LOD BUS REG
000424  012767  120100  176422                  MOV     #<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2 ;SET ATTN CODE READY
000432  016701  176422          1$:             MOV     QR$CR2,R1               ;READ CSR2
000436  032701  000100                          BIT     #Q$ATTN,R1              ;ATTN CLEAR
000442  001373                                  BNE     1$                      ;NO, READ AGAIN
                                        ;
000444  012767  000041  176424                  MOV     #041,QR$LBR             ;CD MEMORY START ADDRESS  =X'21'
000452  012767  120040  176422                  MOV     #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
000460  016701  176422          2$:             MOV     QR$CR2,R1               ;READ CSR2
000464  032701  000040                          BIT     #Q$CCCP,R1              ;IS CC TO CP CLEAR
000470  001373                                  BNE     2$                      ;NO, READ AGAIN
                                        ;
000472  012767  000001  176424                  MOV     #1,QR$LBR               ;TRANSFER COUNT = 1 WORD
000500  012767  120040  176422                  MOV     #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
000506  016701  176422          3$:             MOV     QR$CR2,R1               ;READ CSR2
000512  032701  000040                          BIT     #Q$CCCP,R1              ;IS CC TO CP CLEAR
000516  001373                                  BNE     3$                      ;NO, READ AGAIN
                                        ;
000520  016667  000002  000000G                 MOV     2(SP),DATA1             ;PUT PAGE VALUE INTO BUFFER
000526  005367  000000G                         DEC     DATA1                   ;MICROCODE REINCREMENTS
000532  012767  000000G 176424                  MOV     #DATA1,QR$LBR           ;CC MEMORY DATA BUFFER
000540  012767  120040  176422                  MOV     #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
                                        ;
                                        ;       WAIT FOR INTERRUPT FROM CP
                                        ;
000546                                          WTSE$S  #EFN.3
                                        ;
000560                                          CLEF$S  #EFN.3
                                        ;
```

```
                                        ;       RE-ARM INTERRUPTS
                                        ;
000572  012767  100400  176422          MOV     #<Q$SM+Q$CHB>,QR$CR2    ;CLEAR INTERRUPT (USE HIT BUFFER INT)
000600  012767  101000  176422          MOV     #<Q$SM+Q$REBK>,QR$CR2   ;RE-ARM
000606  012767  160000  176422          MOV     #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2 ;ENABLE
                                        ;
000614  005046                          CLR     -(SP)                   ;CLEAR NOTHING IN CSR1
000616  012746  176000                  MOV     #Q$NCLK,-(SP)           ;SET NO-CLOCKS
000622                                  CALL    CSR1
000626  005067  176422                  CLR     QR$CR2                  ;SET LOAD MODE
                                        ;
000632  012746  001761                  MOV     #1761,-(SP)             ;START CP AT X'3F1'
000636                                  CALL    SEQCS
000642  005046                          CLR     -(SP)                   ;RESET BR INHIBIT
000644                                  CALL    CPCR
000650  012746  000377                  MOV     #377,-(SP)              ;SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
000654                                  CALL    SEQMM
000660  005046                          CLR     -(SP)                   ;RESET BR INHIBIT
000662                                  CALL    MRPCR
000666  012767  001000  176422          MOV     #Q$REBK,QR$CR2          ;RE-ARM INTERRUPTS
000674  012767  120000  176422          MOV     #<Q$SM+Q$ENOP>,QR$CR2   ;SET SEARCH MODE + ENABLE INTERRUPTS
000702  012746  000360                  MOV     #Q$CSEL,-(SP)           ;CLEAR ALL SELECTIONS
000706  052716  001001                  BIS     #<Q$LBD+Q$LBP>,(SP)     ;CLEAR DRIVE AND PULSE
000712  052716  036000                  BIS     #<Q$RNC+Q$CNC>,(SP)     ;CLEAR CP AND PPS NO-CLOCKS
000716  005046                          CLR     -(SP)                   ;SET NOTHING
000720                                  CALL    CSR1
                                        ;
                                        ;       WAIT FOR A TIME
                                        ;
000724  012701  000144                  MOV     #100.,R1                ;LOOP 100 TIMES
000730  005002                          CLR     R2                      ;ADD NOTHING
000732  060202                  4$:     ADD     R2,R2
000734  005301                          DEC     R1
000736  001375                          BNE     4$
                                        ;
000740  005046                          CLR     -(SP)                   ;CLEAR NOTHING IN CSR1
000742  012746  176000                  MOV     #Q$NCLK,-(SP)           ;SET NO-CLOCKS
000746                                  CALL    CSR1
000752  005067  176422                  CLR     QR$CR2                  ;SET LOAD MODE
                                        ;
000756  011666  000002                  MOV     (SP),2(SP)              ;MOVE RETURN ADDRESS DOWN STACK
000762  005726                          TST     (SP)+                   ;POINT TO RETURN ADDRESS
000764                                  RETURN
                                        ;
        000001                          .END
```

```
ALUCKE= 040000
ALUOE = 004000
A01   = 010000
BITVAL= 000000
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BYTE0 = 000000
BYTE1 = 000001
BYTE10= 000012
BYTE11= 000013
BYTE12= 000014
BYTE13= 000015
BYTE14= 000016
BYTE15= 000017
BYTE16= 000020
BYTE17= 000021
BYTE18= 000022
BYTE19= 000023
BYTE2 = 000002
BYTE20= 000024
BYTE21= 000025
BYTE22= 000026
BYTE23= 000027
BYTE24= 000030
BYTE25= 000031
BYTE26= 000032
BYTE27= 000033
BYTE28= 000034
BYTE29= 000035
BYTE3 = 000003
BYTE30= 000036
BYTE31= 000037
BYTE32= 000040
BYTE33= 000041
BYTE34= 000042
BYTE35= 000043
BYTE36= 000044
BYTE37= 000045
BYTE38= 000046
BYTE39= 000047
BYTE4 = 000004
BYTE40= 000050
BYTE41= 000051
BYTE42= 000052
BYTE43= 000053
BYTE44= 000054
BYTE45= 000055
BYTE46= 000056
BYTE47= 000057
BYTE48= 000060
BYTE49= 000061
BYTE5 = 000005
BYTE50= 000062
BYTE51= 000063
BYTE52= 000064
BYTE53= 000065
BYTE54= 000066
BYTE55= 000067
BYTE56= 000070
BYTE57= 000071
BYTE58= 000072
BYTE59= 000073
BYTE6 = 000006
BYTE60= 000074
BYTE61= 000075
BYTE62= 000076
BYTE63= 000077
BYTE64= 000100
BYTE65= 000101
BYTE66= 000102
BYTE67= 000103
BYTE68= 000104
BYTE69= 000105
BYTE7 = 000007
BYTE70= 000106
BYTE71= 000107
BYTE72= 000110
BYTE73= 000111
BYTE74= 000112
BYTE75= 000113
BYTE76= 000114
BYTE77= 000115
BYTE78= 000116
BYTE79= 000117
BYTE8 = 000010
BYTE80= 000120
BYTE81= 000121
BYTE82= 000122
BYTE83= 000123
BYTE84= 000124
BYTE85= 000125
BYTE86= 000126
BYTE87= 000127
BYTE88= 000130
BYTE89= 000131
BYTE9 = 000011
BYTE90= 000132
BYTE91= 000133
BYTE92= 000134
BYTE93= 000135
BYTE94= 000136
BYTE95= 000137
BYTE96= 000140
BYTE97= 000141
BYTE98= 000142
BYTE99= 000143
BYTVAL= 000144
CBKALL= 001000
CBKCLK= 000400
CNOBRE= 100000
CPCCEN= 010000
CPCR  = ****** GX
CPREAD= 040000
CPWRTE= 020000
CSADRD= 000004
CSEQCI= 100000
CSOE  = 000040
CSR1  = ****** GX
CSWRTE= 000100
DATA1 = ****** GX
DBR.RD= 000001
DB$CPP= 001457
DB$SPT= 000026
DB$TPC= 000023
DISPGS= 100000
DMAAWR= 000005
DMARRD= 000003
DMARWR= 000004
EFN.3 = 000003
ENBR  = 010000
LBPP    000000RG    002
LBPSC   000076RG    002
LOC.EN= 000100
LOC.WA= 040000
LOC.WB= 100000
MAREN1= 000001
MAREN2= 004000
MARLOD= 010000
MAROUT= 000002
MAR.LO= 002000
MAR.OU= 000040
MBKALL= 001000
MBKCLK= 000400
MMADRD= 000100
MMLEFT= 000002
MMOE  = 000004
MMWRTE= 000010
MNOBRE= 100000
MREN1 = 000001
MREN2 = 020000
MRPCR = ****** GX
MSYN  = 000040
N     = 000144
PLB   = 000010
PLC   = 000020
PLD   = 000030
PLRWR = 000200
PLR.EN= 000200
PPCR    000244RG    002
PPLB    000174RG    002
QR$CR1= 176420
QR$CR2= 176422
QR$LBR= 176424
Q$ATTN= 000100
Q$BCL = 000001
Q$CCCP= 000040
Q$CHB = 000400
Q$CHRL= 000200
Q$CLR = 000040
Q$CNC = 030000
Q$CP  = 000060
Q$CPCC= 000010
Q$CP2 = 000260
Q$CSC = 010000
Q$CSEL= 000360
Q$CSET= 000002
Q$CSP = 020000
Q$DMA = 000001
Q$ENBK= 040000
Q$ENOP= 020000
Q$FAL = 004000
Q$FC  = 000045
Q$FO  = 000044
Q$FP  = 000046
Q$HBF = 000002
Q$ICP = 000006
Q$IHB = 000003
Q$IHRL= 000002
Q$IMRP= 000007
Q$LBD = 001000
Q$LBDP= 001001
Q$LBP = 000001
Q$LDCD= 000003
Q$LDMD= 000004
Q$LDPP= 002000
Q$LHP = 010000
Q$MNC = 140000
Q$MR  = 000052
Q$MRP = 000040
Q$MRP2= 000240
Q$MSC = 040000
Q$MSET= 000004
Q$MSP = 100000
Q$NCLK= 176000
Q$PP  = 000100
Q$PPSW= 000320
Q$PP2 = 000300
Q$QHLT= 000013
Q$QL  = 000043
Q$QLA = 000053
Q$QLB = 000054
Q$QLR = 000001
Q$QW  = 000042
Q$RDCD= 000005
Q$RDMD= 000006
Q$REBK= 001000
Q$RNC = 006000
Q$RSC = 004000
Q$RSET= 000010
Q$SM  = 100000
Q$SP  = 000120
Q$SP2 = 000340
RGO.EN= 000200
RGO.VA= 020000
SELPG   000326RG    002
SEQCS = ****** GX
SEQMM = ****** GX
SEQ.CI= 000040
S$CLR = 000000
S$LA  = 000001
S$QB  = 000005
S$QR  = 000006
S$QX  = 000004
S$SR  = 000007
S$S1  = 000010
S$S2  = 000014
TD$CTR= 176370
TD$CTW= 176360
TD$INL= 004000
TD$MEM= 000270
TD$OAR= 176344
TD$OTR= 176346
TD$QRD= 000274
TD$SW = 176376
TD$TAR= 176372
TD$TAW= 176362
TD$TDR= 176374
TD$TDW= 176364
T$AD  = 000020
T$BA  = 000002
T$BD  = 000010
T$BSQ = 100000
T$BT  = 000020
T$BTAR= 000030
T$BTD = 002000
T$CD  = 000100
T$CLK = 002000
T$DISK= 000200
T$DRD = 000004
T$EMEM= 010000
T$FSAA= 000000
T$FSAB= 000004
T$FSAC= 000014
T$FSB2= 000010
T$IB  = 000026
T$IBAR= 000024
T$IBE = 020000
T$IBF = 040000
T$ICD = 000040
T$MODE= 004000
T$OB  = 000036
```

SYMBOL TABLE


```
T$0BE = 004000          WORD15= 000036          WORD37= 000112          WORD59= 000166          WORD80= 000240
T$0BF = 010000          WORD16= 000040          WORD38= 000114          WORD6 = 000014          WORD81= 000242
T$0BRA= 000034          WORD17= 000042          WORD39= 000116          WORD60= 000170          WORD82= 000244
T$0BWA= 000032          WORD18= 000044          WORD4 = 000010          WORD61= 000172          WORD83= 000246
T$0UTA= 100000          WORD19= 000046          WORD40= 000120          WORD62= 000174          WORD84= 000250
T$RBDO= 000200          WORD2 = 000004          WORD41= 000122          WORD63= 000176          WORD85= 000252
T$RNB = 000040          WORD20= 000050          WORD42= 000124          WORD64= 000200          WORD86= 000254
T$RSET= 040000          WORD21= 000052          WORD43= 000126          WORD65= 000202          WORD87= 000256
T$SC  = 000022          WORD22= 000054          WORD44= 000130          WORD66= 000204          WORD88= 000260
T$SCLK= 020000          WORD23= 000056          WORD45= 000132          WORD67= 000206          WORD89= 000262
T$SEG1= 000000          WORD24= 000060          WORD46= 000134          WORD68= 000210          WORD9 = 000022
T$SEG2= 000001          WORD25= 000062          WORD47= 000136          WORD69= 000212          WORD90= 000264
T$SEG3= 000002          WORD26= 000064          WORD48= 000140          WORD7 = 000016          WORD91= 000266
T$SO  = 000001          WORD27= 000066          WORD49= 000142          WORD70= 000214          WORD92= 000270
T$UBUS= 100000          WORD28= 000070          WORD5 = 000012          WORD71= 000216          WORD93= 000272
T$1CLK= 000400          WORD29= 000072          WORD50= 000144          WORD72= 000220          WORD94= 000274
T$8BEN= 000020          WORD3 = 000006          WORD51= 000146          WORD73= 000222          WORD95= 000276
UBD.IN= 000020          WORD30= 000074          WORD52= 000150          WORD74= 000224          WORD96= 000300
WORD0 = 000000          WORD31= 000076          WORD53= 000152          WORD75= 000226          WORD97= 000302
WORD1 = 000002          WORD32= 000100          WORD54= 000154          WORD76= 000230          WORD98= 000304
WORD10= 000024          WORD33= 000102          WORD55= 000156          WORD77= 000232          WORD99= 000306
WORD11= 000026          WORD34= 000104          WORD56= 000160          WORD78= 000234          WRDVAL= 000310
WORD12= 000030          WORD35= 000106          WORD57= 000162          WORD79= 000236          XTREAD= 001000
WORD13= 000032          WORD36= 000110          WORD58= 000164          WORD8 = 000020          XTWRTE= 000400
WORD14= 000034

. ABS.  000000     000
        000000     001
PPSUB   000766     002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3288 WORDS  ( 13 PAGES)
DYNAMIC MEMORY:  3860 WORDS  ( 14 PAGES)
ELAPSED TIME:  00:00:46
PPSUB,PPSUB/-SP=[20,1]IM,[20,1]PPSUB
```

```
                                .TITLE MRP
000000                          .PSECT MRP
                                .LIST  MEB

                        ;
                        ;
                        ;       HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
                        ;       MATCH REPORT PROCESSOR ROUTINES
                        ;
                        ;       PARSE THE COMMAND LINE AND TRANSFER CONTROL TO ONE OF THE
                        ;       MRP SUB-MODULES OR BACK TO QMAIN.  CONTROL COULD HAVE BEEN
                        ;       PASSED TO MRP IN ONE OF THREE WAYS:
                        ;       1. FROM QMAIN IF MRP WAS SELECTED AS THE FIRST PROCESSOR
                        ;          UPON ENTRY TO THE PROGRAM.
                        ;          >MR LD MD 0
                        ;       2. FROM QMAIN IF MRP WAS SELECTED FROM ONE OF QMAIN'S
                        ;          OTHER SUB-MODULES.
                        ;          CP>MR LD MD 0
                        ;       3. ON RETURN FROM ONE OF MRP'S SUB-MODULES
                        ;          MR>LD MD 0
                        ;
                        ;       MRP PARSES AT THE SECOND LEVEL OF CONTROL (SEE NOTES AT
                        ;       QMAIN ON LEVELS OF CONTROL).  IN THE EXAMPLES ABOVE, MRP
                        ;       WOULD PARSE 'LD' AND TRANSFER CONTROL TO THE MRP SUB-
                        ;       MODULE MRLD.  IF MRP SHOULD ENCOUNTER A STRING WHICH IS
                        ;       NOT A VALID COMMAND MNEMONIC, MRP CONSIDERS THE STRING TO
                        ;       BE A PROCESSOR MNEMONIC AND RETURNS CONTROL TO QMAIN.  EG:
                        ;       MR>CP LD CS 0
                        ;       THE STRING 'CP' IS NOT A VALID MRP (SECOND LEVEL) COMMAND,
                        ;       MRP RETURNS CONTROL TO QMAIN WHICH IN TURN WILL TRANSFER
                        ;       CONTROL TO ITS SUB-MODULE CP.
                        ;
                        ;       MRP SUB-MODULES:
                        ;       MRLD    LOAD MEMORIES
                        ;       MRPR    PRINT MEMORY CONTENTS
                        ;       MRREST  REMAINING COMMANDS (EXCEPT DEBUGGING)
                        ;       MRBUG   DEBUGGING COMMANDS
```

```
                                        ;
                                        ;
                                        ;       TABLE OF MRP COMMAND MNEMONICS AND THEIR ASSOCIATED ROUTINE
                                        ;       ADDRESSES
                                        ;
                                        ;
000000                                  MRTBL:
000000      101     124                         .ASCII  /AT/            ;SET BREAKPOINT
000002  000000G                                 .WORD   ATi
000004      117     106                         .ASCII  /OF/            ;DISABLE BREAKPOINT
000006  000000G                                 .WORD   OFi
000010      123     123                         .ASCII  /SS/            ;SINGLE STEP
000012  000000G                                 .WORD   SSi
000014      107     117                         .ASCII  /GO/            ;PROCEED FROM BREAKPOINT
000016  000000G                                 .WORD   GOi
000020      123     124                         .ASCII  /ST/            ;STORE INTO A REG
000022  000000G                                 .WORD   STi
000024      122     105                         .ASCII  /RE/            ;READ FROM A REG
000026  000000G                                 .WORD   REi
000030      114     104                         .ASCII  /LD/            ;LOAD MEMORY
000032  000000G                                 .WORD   LDi
000034      120     122                         .ASCII  /PR/            ;PRINT FROM MEMORY
000036  000000G                                 .WORD   PRi
000040      122     123                         .ASCII  /RS/            ;RESET
000042  000000G                                 .WORD   RSi
000044      103     114                         .ASCII  /CL/            ;CALL QR LOADER
000046  000000G                                 .WORD   CLi
        000012                          MRNUM   =       <.-MRTBL>/4
```

```
                                        ;
                                        ;
                                        ;       MRP ROUTINES
                                        ;
                                        ;
000050                                  MRP::
000050  004767  000000G                         JSR     PC,FIND         ;LOCATE THE COMMAND IN THE COMMAND LINE
000054  103003                                  BCC     1$              ;OK, CONTINUE
000056  004767  000000G                         JSR     PC,ERR4
000062  000422                                 BR      MRPXX
000064  022700  000002          1$:     CMP     #2,R0           ;COMMANDS ARE 2 CHARS
000070  001403                                  BEQ     2$
000072  004767  000000G                         JSR     PC,ERR8         ;INCORRECT CHAR COUNT
000076  000414                                  BR      MRPXX           ;TRY AGAIN
                                        ;
000100  012700  000012          2$:     MOV     #MRNUM,R0       ;R0 = NUMBER OF COMMANDS
000104  012702  000000'                 MOV     #MRTBL,R2       ;R2 -> TABLE OF COMMAND MNEMONICS
000110  004767  000000G                 JSR     PC,SCAN         ;FIND MATCH IN TABLE
000114  103003                                  BCC     3$              ;OK, CONTINUE
000116  004767  000000G                         JSR     PC,ERR12        ;COMMAND NOT IN TABLE
000122  000402                                  BR      MRPXX           ;TRY AGAIN
                                        ;
                                        ;       JUMP TO THE ROUTINE THAT GOVERNS THE COMMAND
                                        ;
000124  000171  000000          3$:     JMP     @(R1)
                                        ;
                                        ;
                                        ;       LOCAL MRP LOOP, INCLUDE PROMPT FOR MRP COMMAND,
                                        ;
                                        ;
000130                                  MRPXX::
000130  012767  051115  000000C                 MOV     #"MR,GCMBLK+G.DPRM+2 ;MOVE MRP NAME TO GCM BLOCK
000136  004767  000000G                 JSR     PC,GCONLY       ;PROMPT
000142  004767  000000G                 JSR     PC,FIND         ;LOCATE THE COMMAND IN THE COMMAND LINE
000146  103003                                  BCC     1$              ;OK, CONTINUE
000150  004767  000000G                         JSR     PC,ERR4
000154  000765                                  BR      MRPXX
000156  022700  000002          1$:     CMP     #2,R0           ;COMMANDS ARE 2 CHARS
000162  001403                                  BEQ     2$
000164  004767  000000G                         JSR     PC,ERR8         ;INCORRECT CHAR COUNT
000170  000757                                  BR      MRPXX           ;TRY AGAIN
                                        ;
000172  012700  000012          2$:     MOV     #MRNUM,R0       ;R0 = NUMBER OF COMMANDS
000176  012702  000000'                 MOV     #MRTBL,R2       ;R2 -> TABLE OF COMMAND MNEMONICS
000202  004767  000000G                 JSR     PC,SCAN         ;FIND MATCH IN TABLE
000206  103005                                  BCC     3$              ;OK, CONTINUE
000210  042767  000000G 000000G                 BIC     #BREAK,BASE     ;CLEAR BREAKPOINT FLAG
000216  000167  000000G                         JMP     COMXX           ;RETURN TO QMAIN, LOOK FOR PROCESSOR MNEMONIC
                                        ;
                                        ;       JUMP TO THE ROUTINE THAT GOVERNS THE COMMAND
                                        ;
000222  000171  000000          3$:     JMP     @(R1)
                                        ;
        000001                          .END
```

SYMBOL TABLE


```
ALUCKE= 040000          BYTE4  = 000004          BYTE91= 000133          MREN2  = 020000          Q$NCLK= 176000
ALUOE  = 004000          BYTE40= 000050          BYTE92= 000134          MRNUM  = 000012          Q$PP   = 000100
AT1    = ****** GX       BYTE41= 000051          BYTE93= 000135          MRP      000050RG    002   Q$PPSW= 000320
A01    = 010000          BYTE42= 000052          BYTE94= 000136          MRPXX    000130RG    002   Q$PP2  = 000300
BASE   = ****** GX       BYTE43= 000053          BYTE95= 000137          MRTBL    000000R     002   Q$QHLT= 000013
BITVAL= 000000          BYTE44= 000054          BYTE96= 000140          MSYN   = 000040          Q$QL   = 000043
BIT0   = 000001          BYTE45= 000055          BYTE97= 000141          N      = 000144          Q$QLA  = 000053
BIT1   = 000002          BYTE46= 000056          BYTE98= 000142          OF1    = ****** GX       Q$QLB  = 000054
BIT10  = 002000          BYTE47= 000057          BYTE99= 000143          PLB    = 000010          Q$QLR  = 000001
BIT11  = 004000          BYTE48= 000060          BYTVAL= 000144          PLC    = 000020          Q$QW   = 000042
BIT12  = 010000          BYTE49= 000061          CBKALL= 001000          PLD    = 000030          Q$RDCD= 000005
BIT13  = 020000          BYTE5  = 000005         CBKCLK= 000400          PLRWR  = 000200          Q$RDMD= 000006
BIT14  = 040000          BYTE50= 000062          CL1    = ****** GX       PLR.EN= 000200          Q$REBK= 001000
BIT15  = 100000          BYTE51= 000063          CNOBRE= 100000          PR1    = ****** GX       Q$RNC  = 006000
BIT2   = 000004          BYTE52= 000064          COMXX  = ****** GX       QR$CR1= 176420          Q$RSC  = 004000
BIT3   = 000010          BYTE53= 000065          CPCCEN= 010000          QR$CR2= 176422          Q$RSET= 000010
BIT4   = 000020          BYTE54= 000066          CPREAD= 040000          QR$LBR= 176424          Q$SM   = 100000
BIT5   = 000040          BYTE55= 000067          CPWRTE= 020000          Q$ATTN= 000100          Q$SP   = 000120
BIT6   = 000100          BYTE56= 000070          CSADRD= 000004          Q$BCL  = 000001          Q$SP2  = 000340
BIT7   = 000200          BYTE57= 000071          CSEQCI= 100000          Q$CCCP= 000040          RE1    = ****** GX
BIT8   = 000400          BYTE58= 000072          CSOE   = 000040          Q$CHB  = 000400          RGO.EN= 000200
BIT9   = 001000          BYTE59= 000073          CSWRTE= 000100          Q$CHRL= 000200          RGO.VA= 020000
BREAK  = ****** GX       BYTE6  = 000006         DBR.RD= 000001          Q$CLR  = 000040          R51    = ****** GX
BYTE0  = 000000          BYTE60= 000074          DB$CPP= 001457          Q$CNC  = 030000          SCAN   = ****** GX
BYTE1  = 000001          BYTE61= 000075          DB$SPT= 000026          Q$CP   = 000060          SEQ.CI= 000010
BYTE10= 000012          BYTE62= 000076          DB$TPC= 000023          Q$CPCC= 000010          SS1    = ****** GX
BYTE11= 000013          BYTE63= 000077          DISPGS= 100000          Q$CP2  = 000260          ST1    = ****** GX
BYTE12= 000014          BYTE64= 000100          DMAAWR= 000005          Q$CSC  = 010000          S$CLR  = 000000
BYTE13= 000015          BYTE65= 000101          DMARRD= 000003          Q$CSEL= 000360          S$LA   = 000001
BYTE14= 000016          BYTE66= 000102          DMARWR= 000004          Q$CSET= 000002          S$QB   = 000005
BYTE15= 000017          BYTE67= 000103          ENBR   = 010000          Q$CSP  = 020000          S$QR   = 000006
BYTE16= 000020          BYTE68= 000104          ERR12  = ****** GX       Q$DMA  = 000001          S$QX   = 000004
BYTE17= 000021          BYTE69= 000105          ERR4   = ****** GX       Q$ENBK= 040000          S$SR   = 000007
BYTE18= 000022          BYTE7  = 000007         ERR8   = ****** GX       Q$ENOP= 020000          S$S1   = 000010
BYTE19= 000023          BYTE70= 000106          FIND   = ****** GX       Q$FAL  = 004000          S$S2   = 000014
BYTE2  = 000002          BYTE71= 000107          GCMBLK= ****** GX       Q$FC   = 000045          TD$CTR= 176370
BYTE20= 000024          BYTE72= 000110          GCONLY= ****** GX       Q$FO   = 000044          TD$CTW= 176360
BYTE21= 000025          BYTE73= 000111          GO1    = ****** GX       Q$FP   = 000046          TD$INL= 004000
BYTE22= 000026          BYTE74= 000112          G.DPRM= ****** GX       Q$HBF  = 000002          TD$MEM= 000270
BYTE23= 000027          BYTE75= 000113          LD1    = ****** GX       Q$ICP  = 000006          TD$OAR= 176344
BYTE24= 000030          BYTE76= 000114          LOC.EN= 000100          Q$IHB  = 000003          TD$OTR= 176346
BYTE25= 000031          BYTE77= 000115          LOC.WA= 040000          Q$IHRL= 000002          TD$QRD= 000274
BYTE26= 000032          BYTE78= 000116          LOC.WB= 100000          Q$IMRP= 000007          TD$SW  = 176376
BYTE27= 000033          BYTE79= 000117          MAREN1= 000001          Q$LBD  = 001000          TD$TAR= 176372
BYTE28= 000034          BYTE8  = 000010         MAREN2= 004000          Q$LBDP= 001001          TD$TAW= 176362
BYTE29= 000035          BYTE80= 000120          MARLOD= 010000          Q$LBP  = 000001          TD$TDR= 176374
BYTE3  = 000003          BYTE81= 000121          MAROUT= 000002          Q$LDCD= 000003          TD$TDW= 176364
BYTE30= 000036          BYTE82= 000122          MAR.LO= 002000          Q$LDMD= 000004          T$AD   = 000020
BYTE31= 000037          BYTE83= 000123          MAR.OU= 000040          Q$LDPP= 002000          T$BA   = 000002
BYTE32= 000040          BYTE84= 000124          MBKALL= 001000          Q$LHP  = 010000          T$BD   = 000010
BYTE33= 000041          BYTE85= 000125          MBKCLK= 000400          Q$MNC  = 140000          T$BSO  = 100000
BYTE34= 000042          BYTE86= 000126          MMADRD= 000100          Q$MR   = 000052          T$BT   = 000020
BYTE35= 000043          BYTE87= 000127          MMLEFT= 000002          Q$MRP  = 000040          T$BTAR= 000030
BYTE36= 000044          BYTE88= 000130          MMOE   = 000004          Q$MRR3= 000240          T$BTD  = 002000
BYTE37= 000045          BYTE89= 000131          MMWRTE= 000010          Q$MSC  = 040000          T$CD   = 000100
BYTE38= 000046          BYTE9  = 000011         MNOBRE= 100000          Q$MSET= 000004          T$CLK  = 002000
BYTE39= 000047          BYTE90= 000132          MREN1  = 000001          Q$MSP  = 100000          T$DISK= 000200
```

```
MRP-    M[illegible]0 M1110  27-MAR-80 15:11  PAGE 7-2
SYMBOL TA[illegible]
```

T$DRD = 000004	T$UBUS = 100000	WORD3 = 000006	WORD54 = 000154	WORD79 = 000236
T$EMEM = 010000	T$1CLK = 000400	WORD30 = 000074	WORD55 = 000156	WORD8 = 000020
T$FSAA = 000000	T$8BEN = 000020	WORD31 = 000076	WORD56 = 000160	WORD80 = 000240
T$FSAB = 000004	UBD.IN = 000020	WORD32 = 000100	WORD57 = 000162	WORD81 = 000242
T$FSAC = 000014	WORD0 = 000000	WORD33 = 000102	WORD58 = 000164	WORD82 = 000244
T$FSB2 = 000010	WORD1 = 000002	WORD34 = 000104	WORD59 = 000166	WORD83 = 000246
T$IB = 000026	WORD10 = 000024	WORD35 = 000106	WORD6 = 000014	WORD84 = 000250
T$IBAR = 000024	WORD11 = 000026	WORD36 = 000110	WORD60 = 000170	WORD85 = 000252
T$IBE = 020000	WORD12 = 000030	WORD37 = 000112	WORD61 = 000172	WORD86 = 000254
T$IBF = 040000	WORD13 = 000032	WORD38 = 000114	WORD62 = 000174	WORD87 = 000256
T$ICD = 000040	WORD14 = 000034	WORD39 = 000116	WORD63 = 000176	WORD88 = 000260
T$MODE = 004000	WORD15 = 000036	WORD4 = 000010	WORD64 = 000200	WORD89 = 000262
T$OB = 000036	WORD16 = 000040	WORD40 = 000120	WORD65 = 000202	WORD9 = 000022
T$OBE = 004000	WORD17 = 000042	WORD41 = 000122	WORD66 = 000204	WORD90 = 000264
T$OBF = 010000	WORD18 = 000044	WORD42 = 000124	WORD67 = 000206	WORD91 = 000266
T$OBRA = 000034	WORD19 = 000046	WORD43 = 000126	WORD68 = 000210	WORD92 = 000270
T$OBWA = 000032	WORD2 = 000004	WORD44 = 000130	WORD69 = 000212	WORD93 = 000272
T$OUTA = 100000	WORD20 = 000050	WORD45 = 000132	WORD7 = 000016	WORD94 = 000274
T$RBDO = 000200	WORD21 = 000052	WORD46 = 000134	WORD70 = 000214	WORD95 = 000276
T$RNB = 000040	WORD22 = 000054	WORD47 = 000136	WORD71 = 000216	WORD96 = 000300
T$RSET = 040000	WORD23 = 000056	WORD48 = 000140	WORD72 = 000220	WORD97 = 000302
T$SC = 000022	WORD24 = 000060	WORD49 = 000142	WORD73 = 000222	WORD98 = 000304
T$SCLK = 020000	WORD25 = 000062	WORD5 = 000012	WORD74 = 000224	WORD99 = 000306
T$SEG1 = 000000	WORD26 = 000064	WORD50 = 000144	WORD75 = 000226	WRDVAL = 000310
T$SEG2 = 000001	WORD27 = 000066	WORD51 = 000146	WORD76 = 000230	XTREAD = 001000
T$SEG3 = 000002	WORD28 = 000070	WORD52 = 000150	WORD77 = 000232	XTWRTE = 000400
T$SO = 000001	WORD29 = 000072	WORD53 = 000152	WORD78 = 000234	

```
. ABS.  000000    000
        000000    001
MRP.    000226    002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3151 WORDS ( 13 PAGES)
DYNAMIC MEMORY:  3860 WORDS ( 14 PAGES)
ELAPSED TIME:  00:00:41
MRP,MRP/-SP=[20,1]IM,[20,1]MRP
```

```
                                        .TITLE  MRLD
                                        .PSECT  MRLD
                                ;
                                ;
                                ;       HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
                                ;       MATCH REPORT PROCESSOR ROUTINES
                                ;
                                ;       COMMANDS:
                                ;       LD      LOAD MRP MEMORIES
                                ;
                                ;       ONCE A COMMAND HAS BEEN EXECUTED (OR AN ERROR ENCOUNTERED)
                                ;       THIS MODULE RETURNS CONTROL TO THE MODULE MRP AT LOCATION
                                ;       'MRPXX'.
                                ;
                                ;
                                        .MCALL  WTSE$S,RDAF$S,CLEF$S
                                ;
                                ;
                                ;       TABLE OF VALID MRP MEMORY MNEMONICS AND ADDRESSES OF
                                ;       ASSOCIATED ROUTINES
                                ;
000000                          LD1TBL::
000000  000115      115               .ASCII  /MM/                    ;LOAD MICROPGM MEMORY
000002  000262'                      .WORD   LD1MM
000004  000115      104               .ASCII  /MD/                    ;LOAD DATA MEMORY (X AND T TABLES)
000006  000424'                      .WORD   LD1MD
        000002          LD1LN   ==      <.-LD1TBL>/4
                                ;
                                ;
                                ;       LOAD
                                ;       PERFORM THIRD LEVEL PARSING
                                ;       EG. IN THE COMMAND:
                                ;       MR>LD MM 0
                                ;       PARSE THE 'MM'
                                ;
                                ;
000010                          LD1::
000010                                  CALL    FIND                    ;LOCATE MEMORY MNEMONIC IN COMMAND LINE
000014  103004                          BCC     1$                      ;OK, CONTINUE
000016                                  CALL    ERR4                    ;'MISSING OPERAND'
000022  000167  000644                  JMP     LD1X                    ;EXIT
                                ;
                                ;       MATCH MNEMONIC IN COMMAND LINE AGAINST TABLE OF VALID MNEMONICS
                                ;
000026  012700  000002          1$:     MOV     #LD1LN,R0               ;NUMBER OF TABLE ENTRIES
000032  012702  000000'                  MOV     #LD1TBL,R2              ;POINT TO TABLE
000036                                  CALL    SCAN                    ;MATCH AGAINST COMMAND LINE
000042  103004                          BCC     2$                      ;MATCH WAS MADE
000044                                  CALL    ERR7                    ;'INVALID MEMORY MNEMONIC'
000050  000167  000616                  JMP     LD1X
                                ;
                                ;       SAVE POINTER TO ROUTINE ASSOCIATED WITH THE MEMORY MNEMONIC
                                ;       SCAN THE REMAINDER OF THE COMMAND LINE FOR CONTROL INFORMATION
                                ;
                                ;       1. START ADDRESS ONLY, REPEAT PROMPT FOR EACH MEMORY LOCATION
                                ;          FROM THE START ADDRESS FORWARD, NB, END OF MEMORY (AS DETECTED
                                ;          BY THE QMAIN SUBROUTINE PDATA) OR A <CR> RESPONSE TO THE PROMPT
```

```
                                        ;           WILL·TERMINATE·THE·LOAD.
                                        ;           MR>LD·MD·0
                                        ;       2. START·ADDRESS, LOOP INDICATOR, LOOP·ON·THE·LOADING·OF·
                                        ;           THIS·ONE·MEMORY·LOCATION·ONLY.
                                        ;           MR>LD·MD·0 L·
                                        ;       3. START·ADDRESS, END·ADDRESS, NO·LOOP, ISSUE·ONLY·ONE·
                                        ;           PROMPT·AND·FILL·MEMORY·(BETWEEN·START·AND·END·ADDRESSES)
                                        ;           WITH·THIS VALUE.
                                        ;           MR>LD·MD·0 7
                                        ;       4. START·ADDRESS, END·ADDRESS, LOOP·INDICATOR, ISSUE ONLY·ONE·
                                        ;           PROMPT·AND·LOOP·ON·THE·LOADING·OF·MEMORY·(BETWEEN START·
                                        ;           AND·END·ADDRESSES) WITH·THIS·VALUE.
                                        ;           MR>LD·MD·0 7 L·
                                        ;
                                        ;
000054  010167  000000G·        2$:     MOV·    R1,RTNPT·               ;SAVE·POINTER·
000060                                  CALL·   FIND                    ;LOCATE·START·ADDRESS·IN·COMMAND·LINE·
000064  103004                          BCC·    3$                      ;OK, CONTINUE·
000066                                  CALL·   ERR4                    ;'MISSING·OPERAND'
000072·000167  000574·                  JMP·    LD1X                    ;EXIT·
000076                          3$:     CALL·   PACK                    ;CONVERT·COMMAND·LINE·VALUE·TO BINARY·
000102·103004                           BCC·    4$                      ;CONVERSION·SUCCESSFUL·
000104                                  CALL·   ERR5                    ;'INVALID·NUMERIC·VALUE'
000110  000167  000556                  JMP·    LD1X
                                        ;
000114  016767  000000G·000000G· 4$:     MOV·    BINWD,MSTRT·            ;SAVE·LOADING·START·ADDRESS·
000122·016767  000000G·000000G·         MOV·    BINWD,MSTR2·            ;SAVE·IT·TWICE·(FOR·REFRESH·ON·LOOP)
000130  012767  177777·  000000G·       MOV·    #-1,MEND·               ;INIT·END·ADDRESS·
                                        ;
                                        ;       START·ADDRESS·HAS·BEEN·FOUND.·SCAN·FOR·END ADDRESS·OR·LOOP
                                        ;       INDICATOR·(CONDITIONS·2,3,·OR·4).  IF·THERE·IS·NOTHING·
                                        ;       FURTHER·IN·THE·COMMAND·LINE, CONDITION·1 IS·IN·EFFECT.
                                        ;
000136                                  CALL·   FIND                    ;SCAN·COMMAND·LINE·
000142·103004                           BCC·    5$                      ;SOMETHING THERE·
000144  052767  000000G·000000G·        BIS·    #RP,BASE·               ;SIGNAL·TO REPEAT·PROMPT·
000152·000437                           BR       9$                     ;JUMP·TO·RTN·
                                        ;
000154  122711  000114          5$:     CMPB·   #'L,(R1)                ;LOOP·INDICATOR·
000160  001006                          BNE·    6$                      ;NO, MUST·BE·UPPER·ADDRESS·
000162·016767  000000G·000000G·         MOV·    MSTRT,MEND·             ;SET·END·ADDR·=·START·ADDR·
000170                                  CALL·   HANG                    ;HOW·TO·STOP·LOOP·
000174  000426                          BR      9$                      ;JUMP·TO·RTN·
                                        ;
000176                          6$:     CALL·   PACK                    ;CONVERT·UPPER·ADDRESS·
000202·103004                           BCC·    7$                      ;OK, CONTINUE·
000204                                  CALL·   ERR5                    ;INVALID·NUMERIC·
000210  000167  000456                  JMP·    LD1X                    ;EXIT·
                                        ;
                                        ;       SAVE·END·ADDRESS·(BINARY)
                                        ;       CHECK·FOR·LOOP·INDICATOR·AFTER·END·ADDRESS (CONDITION·4)
                                        ;
000214  016767  000000G·000000G· 7$:     MOV·    BINWD,MEND·             ;SET·UP·END·ADDRESS·
000222·                                 CALL·   FIND                    ;CHECK·FOR·LOOP·INDICATOR·
000226·103411                           BCS·    9$                      ;NO·LOOP·
000230  122711  000114                  CMPB·   #'L,(R1)                ;CORRECT·INDICATOR·
000234  001404                          BEQ·    8$                      ;YES, CONTINUE·
```

```
115 000236                                      CALL    ERR11           ;LOOP OPTION ERROR
116 000242  000167  000424                      JMP     LD1X
117 000246                              8$:     CALL    HANG                    ;HOW TO STOP LOOP
118                                             ;
119 000252  016701  000000G             9$:     MOV     RTNPT,R1                ;POINT TO ROUTINE
120 000256  000171  000000                      JMP     @(R1)                   ;JUMP TO ROUTINE
121                                             ;
122                                             ;
123                                             ;       LOAD MICROPGM MEMORY
124                                             ;
125 000262                              LD1MM::
126 000262  016746  000000G                     MOV     MMHIGH,-(SP)            ;SUPPLY UPPER MEMORY LIMIT
127 000266  016746  000000G                     MOV     MMLOW,-(SP)             ;LOWER LIMIT
128 000272                                      CALL    BUFS2                   ;PREPARE TO LOAD
129 000276  103002                              BCC     100$                    ;NO ERRORS, CONTINUE
130 000300  000167  000366                      JMP     LD1X
131                                             ;
132 000304  032767  000000G 000000G 100$:       BIT     #RP,BASE                ;REPEAT PROMPT
133 000312  001416                              BEQ     2$                      ;NO, ONCE ONLY
134 000314                              1$:     CALL    PDATA                   ;READ DATA FROM COMMAND LINE
135 000320  103002                              BCC     10$                     ;NOT END OF MEMORY
136 000322  000167  000344                      JMP     LD1X                    ;IF END, EXIT
137 000326  102002                      10$:    BVC     20$                     ;<CR> RESPONSE
138 000330  000167  000336                      JMP     LD1X                    ;YES, EXIT
139 000334                              20$:    CALL    LOADMM                  ;LOAD MICRO MEMORY
140 000340  066767  000000G 000000G             ADD     INCVAL,MSTR2            ;BUMP TO NEXT ADDRESS
141 000346  000762                              BR      1$                      ;REPEAT
142                                             ;
143                                             ;       PROMPT ONCE THEN FILL MEMORY
144                                             ;
145 000350                              2$:     CALL    PDATA                   ;PROMPT
146 000354  103002                              BCC     30$                     ;NOT END OF MEMORY OR ERROR
147 000356  000167  000310                      JMP     LD1X                    ;ELSE EXIT
148                                             ;
149 000362                              30$:    CALL    LOADMM                  ;LOAD MICRO MEMORY
150 000366  066767  000000G 000000G             ADD     INCVAL,MSTR2            ;ADVANCE ADDRESS
151 000374  026767  000000G 000000G             CMP     MSTR2,MEND              ;HAS UPPER MEMORY LIMIT BEEN REACHED
152 000402  101767                              BLOS    30$                     ;NO, CONTINUE
153 000404  032767  000000G 000000G             BIT     #LOOP,BASE              ;LOOP ON
154 000412  001527                              BEQ     LD1X                    ;NO, EXIT
155 000414  016767  000000G 000000G             MOV     MSTRT,MSTR2             ;REINITIALIZE ADDRESS
156 000422  000757                              BR      30$
157                                             ;
158                                             ;
159                                             ;       LOAD DATA MEMORY
160                                             ;
161 000424                              LD1MD::
162 000424  016746  000000G                     MOV     MDHIGH,-(SP)            ;SUPPLY UPPER MEMORY LIMIT
163 000430  016746  000000G                     MOV     MDLOW,-(SP)             ;LOWER LIMIT
164 000434                                      CALL    BUFSET                  ;PREPARE FOR LOAD
165 000440  103514                              BCS     LD1X                    ;ERROR
166                                             ;
167 000442  012746  000377                      MOV     #377,-(SP)              ;SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
168 000446                                      CALL    SEQMM
169                                             ;
170 000452  005046                              CLR     -(SP)                   ;REINHIBIT BRANCH CONTROL REG
171 000454                                      CALL    MRPCR
```

```
 172·000460  005046                          CLR·     -(SP)
 173·000462·                                  CALL·    LBMRP·                ;SINGLE·CLOCK·TO·REINHIBIT·BRANCH·REGISTER·
 174                                  ;
 175·000466  005046                          CLR·     -(SP)                 ;START·MICROCODE·AT·0
 176·000470                                   CALL·    SEQCS·
 177                                  ;
 178·000474  005046                          CLR·     -(SP)                 ;REINHIBIT BRANCH·CONTROL·REG·
 179·000476                                   CALL·    CPCR
 180·000502·  005046                         CLR·     -(SP)
 181·000504                                   CALL·    LBCP                  ;SINGLE·CLOCK·TO·REINHIBIT·BRANCH·REGISTER·
 182·                                 ;
 183·000510  012767  001000  176422·         MOV·     #Q$REBK,QR$CR2        ;RE-ARM·INTERRUPTS·
 184·000516  012767  120000  176422·         MOV·     #<Q$SM+Q$ENOP>,QR$CR2·;SET·SEARCH·MODE·+ ENABLE·INTERRUPTS·
 185·000524  012746  000360                  MOV·     #Q$CSEL,-(SP)         ;CLEAR·ALL SELECTIONS·
 186·000530  052716  001001                  BIS·     #<Q$LBD+Q$LBP>,(SP)   ;CLEAR·DRIVE·AND·PULSE·
 187·000534  052716  170000                  BIS·     #<Q$MNC+Q$CNC>,(SP)   ;CLEAR·CP·NO-CLOCK·
 188·000540  005046                          CLR·     -(SP)                 ;SET·NOTHING·
 189·000542·                                  CALL·    CSR1
 190                                  ;
 191·000546  032767  000000G·000000G·        BIT·     #RP,BASE·             ;REPEAT·PROMPT·
 192·000554  001412·                          BEQ·     2$                    ;NO, ONCE·ONLY·
 193·000556                           1$:     CALL·    PDATA·                ;READ·DATA FROM·COMMAND·LINE·
 194·000562·  103434                          BCS·     LDMDX·                ;EXIT·ON·END·OF·MEMORY·OR·ERROR·
 195·000564  102433                          BVS·     LDMDX·                ;<CR> RESPONSE·TO·PROMPT·
 196·000566                                   CALL·    LOADMD·               ;LOAD·MRP·DATA·MEMORY·
 197·000572·  066767  000000G·000000G·        ADD·     INCVAL,MSTR2·         ;BUMP·ADDRESS·
 198·000600  000766                          BR       1$                    ;REPEAT·
 199                                  ;
 200                                  ;       PROMPT·ONCE·THEN·FILL·MEMORY·
 201                                  ;
 202·000602·                          2$:     CALL·    PDATA·                ;PROMPT·
 203·000606  103422·                          BCS·     LDMDX·                ;END·OF·MEMORY·OR·ERROR·
 204·000610  102421                          BVS·     LDMDX·                ;<CR> RESPONSE·
 205·000612·                          3$:     CALL·    LOADMD·               ;LOAD·DATA MEMORY·
 206·000616  066767  000000G·000000G·        ADD·     INCVAL,MSTR2·         ;ADVANCE·ADDRESS·
 207·000624  026767  000000G·000000G·        CMP·     MSTR2,MEND·           ;HAS·UPPER MEMORY·LIMIT·BEEN·REACHED·
 208·000632·  101767                          BLOS·    3$                    ;NO, CONTINUE·
 209·000634  032767  000000G·000000G·        BIT·     #LOOP,BASE·           ;LOOP·ON·
 210·000642·  001404                          BEQ·     LDMDX·                ;EXIT·
 211·000644  016767  000000G·000000G·        MOV·     MSTRT,MSTR2·          ;REINITIALIZE·ADDRESS·
 212·000652·  000757                          BR       3$
 213                                  ;
 214·000654  005046                  LDMDX:  CLR·     -(SP)                 ;CLEAR·NOTHING·IN·CSR1
 215·000656  012746  176000·                  MOV·     #Q$NCLK,-(SP)         ;SET·NO-CLOCKS·
 216·000662·                                  CALL·    CSR1
 217·000666·  005067  176422                  CLR·     QR$CR2·               ;SET·LOAD·MODE·
 218                                  ;
 219                                  ;
 220·000672·                          LD1X:
 221·000672·  042767  000000G·000000G·        BIC·     #RP,BASE·             ;CLEAR·PROMPT·FLAG·
 222·000700                                   CALL·    KILL                  ;KILL·AST·(IF·THERE·WAS·ONE)
 223·000704  000167  000000G·                 JMP·     MRPXX·
```

```
                                        ;
                                        ;
                                        ;       LOAD·MRP·MICROPGM·MEMORY.
                                        ;
                                        ;
000710                                  LOADMM:
000710  016746  000000G·                        MOV·    MSTR2,-(SP)             ;SEND CURRENT·ADDRESS·
000714                                          CALL·   SEQMM·                  ;SEQUENCE·UP·TO·START·ADDR·
000720  012746  000012                          MOV·    #<MMWRTEN+MMLEFT>,-(SP)
000724                                          CALL·   MRPCRA·                 ;DIRECT·CNTL·WORD·TO·MRP·
000730  016746  000000G·                        MOV·    DATA1,-(SP)             ;MOVE·DATA·WORD·TO·LOD·BUS·REG
000734                                          CALL·   LBMSC·                  ;SEND·DATA·WORD·TO·MRP·         NS·
                                        ;
000740  016746  000000G·                        MOV·    MSTR2,-(SP)             ;SEND CURRENT·ADDRESS·
000744                                          CALL·   SEQMM·                  ;SEQUENCE·UP·TO·START·ADDRESS·
000750  012746  000010                          MOV·    #<MMWRTEN>,-(SP)
000754                                          CALL·   MRPCRA·                 ;DIRECT·CNTL·WORD·TO·MRP·
000760  016746  000000G·                        MOV·    DATA2,-(SP)             ;MOVE·DATA·WORD·TO·LOD·BUS·REG
000764                                          CALL·   LBMSC·                  ;SEND·DATA·WORD·TO·MRP·         NS·
                                        ;
000770  005046                                  CLR·    -(SP)
000772·                                         CALL·   MRPCR·                  ; CLEAR·THE·MRP·CR·BITS·        NS·
000776  005046                                  CLR·    -(SP)
001000                                          CALL·   LBMRP·                  ;ISSUE·CLOCK·TO·REINHIBIT·BR·
001004                                          RETURN·
```

```
                                        ;
                                        ;
                                        ;       LOAD·MRP·DATA·MEMORY·
                                        ;
                                        ;
001006                                  LOADMD:
001006  012767  000004  176424                  MOV·    #Q$LDMD,QR$LBR          ;MOVE·ATTN CODE·TO·LOD·BUS·REG
001014  012767  120100  176422·                 MOV·    #<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2·;SET·ATTN·CODE·READY
001022  016701  176422                  1$:     MOV·    QR$CR2,R1               ;READ·CSR2
001026  032701  000100                          BIT·    #Q$ATTN,R1              ;ATTN·CLEAR·
001032· 001373                                  BNE·    1$                      ;NO, READ·AGAIN·
                                        ;
001034  016767  000000G·176424                  MOV·    MSTR2,QR$LBR·           ;CD MEMORY START ADDRESS
001042· 012767  120040  176422·                 MOV·    #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2·;SET·CC·TO·CP·
001050  016701  176422                  2$:     MOV·    QR$CR2,R1               ;READ CSR2
001054  032701  000040                          BIT·    #Q$CCCP,R1              ;IS·CC·TO·CP·CLEAR·
001060  001373                                  BNE·    2$                      ;NO, READ·AGAIN·
                                        ;
001062· 012767  000001  176424                  MOV·    #1,QR$LBR·              ;TRANSFER·COUNT·=·1 WORD·
001070  012767  120040  176422·                 MOV·    #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2·;SET·CC·TO·CP·
001076  016701  176422                  3$:     MOV·    QR$CR2,R1               ;READ CSR2
001102· 032701  000040                          BIT·    #Q$CCCP,R1              ;IS·CC·TO·CP·CLEAR·
001106  001373                                  BNE·    3$                      ;NO, READ·AGAIN·
                                        ;
001110  012767  000000G·176424                  MOV·    #DATA1,QR$LBR·          ;CC·MEMORY DATA·BUFFER·
001116  012767  120040  176422·                 MOV·    #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2·;SET·CC·TO·CP·
                                        ;
                                        ;       WAIT·FOR·INTERRUPT·FROM·CP·
                                        ;
001124                                          WTSE$S·  #EFN.3
                                        ;
001136                                          CLEF$S·  #EFN.3
                                        ;
                                        ;       RE-ARM·INTERRUPTS·
                                        ;
001150  012767  100400  176422·                 MOV·    #<Q$SM+Q$CHB>,QR$CR2·;CLEAR·INTERRUPT·(USE·HIT·BUFFER·INT)
001156  012767  101000  176422·                 MOV·    #<Q$SM+Q$REBK>,QR$CR2·;RE-ARM·
001164  012767  160000  176422·                 MOV·    #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2·;ENABLE·
001172·                                         RETURN·
                                        ;
000001                                          .END·
```

```
ALUCKE= 040000
ALUOE = 004000
A01   = 010000
BASE  = ****** GX
BINWD = ****** GX
BITVAL= 000000
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BUFSET= ****** GX
BUFS2 = ****** GX
BYTE0 = 000000
BYTE1 = 000001
BYTE10= 000012
BYTE11= 000013
BYTE12= 000014
BYTE13= 000015
BYTE14= 000016
BYTE15= 000017
BYTE16= 000020
BYTE17= 000021
BYTE18= 000022
BYTE19= 000023
BYTE2 = 000002
BYTE20= 000024
BYTE21= 000025
BYTE22= 000026
BYTE23= 000027
BYTE24= 000030
BYTE25= 000031
BYTE26= 000032
BYTE27= 000033
BYTE28= 000034
BYTE29= 000035
BYTE3 = 000003
BYTE30= 000036
BYTE31= 000037
BYTE32= 000040
BYTE33= 000041
BYTE34= 000042
BYTE35= 000043
BYTE36= 000044
BYTE37= 000045
BYTE38= 000046
BYTE39= 000047
BYTE4 = 000004
BYTE40= 000050
BYTE41= 000051
BYTE42= 000052
BYTE43= 000053
BYTE44= 000054
BYTE45= 000055
BYTE46= 000056
BYTE47= 000057
BYTE48= 000060
BYTE49= 000061
BYTE5 = 000005
BYTE50= 000062
BYTE51= 000063
BYTE52= 000064
BYTE53= 000065
BYTE54= 000066
BYTE55= 000067
BYTE56= 000070
BYTE57= 000071
BYTE58= 000072
BYTE59= 000073
BYTE6 = 000006
BYTE60= 000074
BYTE61= 000075
BYTE62= 000076
BYTE63= 000077
BYTE64= 000100
BYTE65= 000101
BYTE66= 000102
BYTE67= 000103
BYTE68= 000104
BYTE69= 000105
BYTE7 = 000007
BYTE70= 000106
BYTE71= 000107
BYTE72= 000110
BYTE73= 000111
BYTE74= 000112
BYTE75= 000113
BYTE76= 000114
BYTE77= 000115
BYTE78= 000116
BYTE79= 000117
BYTE8 = 000010
BYTE80= 000120
BYTE81= 000121
BYTE82= 000122
BYTE83= 000123
BYTE84= 000124
BYTE85= 000125
BYTE86= 000126
BYTE87= 000127
BYTE88= 000130
BYTE89= 000131
BYTE9 = 000011
BYTE90= 000132
BYTE91= 000133
BYTE92= 000134
BYTE93= 000135
BYTE94= 000136
BYTE95= 000137
BYTE96= 000140
BYTE97= 000141
BYTE98= 000142
BYTE99= 000143
BYTVAL= 000144
CBKALL= 001000
CBKCLK= 000400
CNOBRE= 100000
CPCCEN= 010000
CPCR  = ****** GX
CPREAD= 040000
CPWRTE= 020000
CSADRD= 000004
CSEQCI= 100000
CSOE  = 000040
CSR1  = ****** GX
CSWRTE= 000100
DATA1 = ****** GX
DATA2 = ****** GX
DBR.RD= 000001
DB$CPP= 001457
DB$SPT= 000026
DB$TPC= 000023
DISPGS= 100000
DMAAWR= 000005
DMARRD= 000003
DMARWR= 000004
EFN.3 = ****** GX
ENBR  = 010000
ERR11 = ****** GX
ERR4  = ****** GX
ERR5  = ****** GX
ERR7  = ****** GX
FIND  = ****** GX
HANG  = ****** GX
INCVAL= ****** GX
KILL  = ****** GX
LBCP  = ****** GX
LBMRP = ****** GX
LBMSC = ****** GX
LDMDX   000654R
LD1     000010RG
LD1LN = 000002 G
LD1MD   000424RG
LD1MM   000262RG
LD1TBL  000000RG
LD1X    000672R
LOADMD  001006R
LOADMM  000710R
LOC.EN= 000100
LOC.WA= 040000
LOC.WB= 100000
LOOP  = ****** GX
MAREN1= 000001
MAREN2= 004000
MARLOD= 010000
MAROUT= 000002
MAR.LD= 002000
MAR.OU= 000040
MBKALL= 001000
MBKCLK= 000400
MDHIGH= ****** GX
MDLOW = ****** GX
MEND  = ****** GX
MMADRD= 000100
MMHIGH= ****** GX
MMLEFT= 000002
MMLOW = ****** GX
MMOE  = 000004
MMWRTE= 000010
MNOBRE= 100000
MREN1 = 000001
MREN2 = 020000
MRPCR = ****** GX
MRPCRA= ****** GX
MRPXX = ****** GX
MSTRT = ****** GX
MSTR2 = ****** GX
MSYN  = 000040
N     = 000144
PACK  = ****** GX
PDATA = ****** GX
PLB   = 000010
PLC   = 000020
PLD   = 000030
PLRWR = 000200
PLR.EN= 000200
QR$CR1= 176420
QR$CR2= 176422
QR$LBR= 176424
Q$ATTN= 000100
Q$BCL = 000001
Q$CCCP= 000040
Q$CHB = 000400
Q$CHRL= 000200
Q$CLR = 000040
Q$CNC = 030000
002 Q$CP  = 000060
002 Q$CPCC= 000010
    Q$CP2 = 000260
002 Q$CSC = 010000
002 Q$CSEL= 000360
002 Q$CSET= 000002
002 Q$CSP = 020000
002 Q$DMA = 000001
002 Q$ENBK= 040000
    Q$ENOP= 020000
    Q$FAL = 004000
Q$FC  = 000045
Q$FO  = 000044
Q$FP  = 000046
Q$HBF = 000002
Q$ICP = 000006
Q$ITHB= 000003
Q$IHRL= 000002
Q$IMRP= 000007
Q$LBD = 001000
Q$LBDP= 001001
Q$LBP = 000001
Q$LDCD= 000003
Q$LDMD= 000004
Q$LDPP= 002000
Q$LHP = 010000
Q$MNG = 140000
Q$MR  = 000052
Q$MRP = 000040
Q$MRP2= 000240
Q$MSC = 040000
Q$MSET= 000004
Q$MSP = 100000
Q$NCLK= 176000
Q$PP  = 000100
Q$PPSW= 000320
Q$PP2 = 000300
Q$QHLT= 000013
Q$QL  = 000043
Q$QLA = 000053
Q$QLB = 000054
Q$QLR = 000001
Q$QW  = 000042
Q$RDCD= 000005
Q$RDMD= 000006
Q$REBK= 001000
Q$RNC = 006000
Q$RSC = 004000
Q$RSET= 000010
Q$SM  = 100000
Q$SP  = 000120
Q$SP2 = 000340
RGQ.EN= 000200
RGQ.VA= 020000
RP    = ****** GX
RTNPT = ****** GX
SCAN  = ****** GX
SEQCS = ****** GX
SEQMM = ****** GX
SEQ.CI= 000010
S$CLR = 000000
S$LA  = 000001
S$QB  = 000005
S$QR  = 000006
S$QX  = 000004
S$SR  = 000007
S$S1  = 000010
S$S2  = 000014
```

SYMBOL·TABLE·

```
TD$CTR=·176370          T$ICD·=·000040          WORD18=·000044          WORD46=·000134          WORD74=·000224
TD$CTW=·176360          T$MODE=·004000          WORD19=·000046          WORD47=·000136          WORD75=·000226
TD$INL=·004000          T$OB··=·000036          WORD2·=·000004          WORD48=·000140          WORD76=·000230
TD$MEM=·000270          T$OBE·=·004000          WORD20=·000050          WORD49=·000142·         WORD77=·000232
TD$OAR=·176344          T$OBF·=·010000          WORD21=·000052·         WORD5·=·000012·         WORD78=·000234
TD$OTR=·176346          T$OBRA=·000034          WORD22=·000054          WORD50=·000144          WORD79=·000236
TD$ORD=·000274          T$OBWA=·000032·         WORD23=·000056          WORD51=·000146          WORD8·=·000020
TD$SW·=·176376          T$OUTA=·100000          WORD24=·000060          WORD52=·000150          WORD80=·000240
TD$TAR=·176372·         T$RBDO=·000200          WORD25=·000062·         WORD53=·000152·         WORD81=·000242
TD$TAW=·176362·         T$RNB·=·000040          WORD26=·000064          WORD54=·000154          WORD82=·000244
TD$TDR=·176374          T$RSET=·040000          WORD27=·000066          WORD55=·000156          WORD83=·000246
TD$TDW=·176364          T$SC··=·000022·         WORD28=·000070          WORD56=·000160          WORD84=·000250
T$AD··=·000020          T$SCLK=·020000          WORD29=·000072·         WORD57=·000162·         WORD85=·000252
T$BA··=·000002·         T$SEG1=·000000          WORD3·=·000006          WORD58=·000164          WORD86=·000254
T$BD··=·000010          T$SEG2=·000001          WORD30=·000074          WORD59=·000166          WORD87=·000256
T$BSO·=·100000          T$SEG3=·000002·         WORD31=·000076          WORD6·=·000014          WORD88=·000260
T$BT··=·000020          T$SO··=·000001          WORD32=·000100          WORD60=·000170          WORD89=·000262
T$BTAR=·000030          T$UBUS=·100000          WORD33=·000102·         WORD61=·000172·         WORD9·=·000022
T$BTD·=·002000          T$1CLK=·000400          WORD34=·000104          WORD62=·000174          WORD90=·000264
T$CD··=·000100          T$8BEN=·000020          WORD35=·000106          WORD63=·000176          WORD91=·000266
T$CLK·=·002000          UBD.IN=·000020          WORD36=·000110          WORD64=·000200          WORD92=·000270
T$DISK=·000200          WORD0·=·000000          WORD37=·000112·         WORD65=·000202·         WORD93=·000272
T$DRD·=·000004          WORD1·=·000002·         WORD38=·000114          WORD66=·000204          WORD94=·000274
T$EMEM=·010000          WORD10=·000024          WORD39=·000116          WORD67=·000206          WORD95=·000276
T$FSAA=·000000          WORD11=·000026          WORD4·=·000010          WORD68=·000210          WORD96=·000300
T$FSAB=·000004          WORD12=·000030          WORD40=·000120          WORD69=·000212·         WORD97=·000302
T$FSAC=·000014          WORD13=·000032·         WORD41=·000122·         WORD7·=·000016          WORD98=·000304
T$FSB2=·000010          WORD14=·000034          WORD42=·000124          WORD70=·000214          WORD99=·000306
T$IB··=·000026          WORD15=·000036          WORD43=·000126          WORD71=·000216          WRDVAL=·000310
T$IBAR=·000024          WORD16=·000040          WORD44=·000130          WORD72=·000220          XTREAD=·001000
T$IBE·=·020000          WORD17=·000042·         WORD45=·000132·         WORD73=·000222·         XTWRTE=·000400
T$IBF·=·040000

. ABS.  000000     000
        000000     001
MRLD·   001174     002·
ERRORS·DETECTED:  0

VIRTUAL·MEMORY·USED:  3608 WORDS·  (·15 PAGES)
DYNAMIC·MEMORY:  4916 WORDS·  ( 18 PAGES)
ELAPSED·TIME:  00:00:48
MRLD,MRLD/-SP=[20,1]IM,[20,1]MRLD·
```

```
                                        .TITLE  MRPR
000000                                  .PSECT  MRPR
                                ;
                                ;
                                ;       HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
                                ;       MATCH REPORT PROCESSOR ROUTINES
                                ;
                                ;       COMMANDS:
                                ;       PR      PRINT CONTENTS OF MRP MEMORY
                                ;
                                ;       ONCE A COMMAND HAS BEEN EXECUTED (OR AN ERROR ENCOUNTERED)
                                ;       THIS MODULE RETURNS CONTROL TO THE MODULE MRP AT LOCATION
                                ;       'MRPXX'.
                                ;
                                ;
                                ; **********************
                                ;
                                ;       NOTE:
                                ;       THE COMMAND TO PRINT MRP DATA MEMORY CANNOT BE USED
                                ;       UNLESS MRP MICROPROGRAM MEMORY AND CP CONTROL STORE
                                ;       HAVE BEEN LOADED VIA THE 'LOADER' PROGRAM.
                                ;
                                ; **********************
                                ;
                                ;
                                        .MCALL  CLEF$S,WTSE$S,RDAF$S
                                ;
                                ;
                                ;       TABLE OF VALID MRP MEMORY MNEMONICS AND ADDRESSES OF
                                ;       ASSOCIATED ROUTINES
                                ;
000000                          PR1TBL::
000000    115     115                   .ASCII  /MM/                    ;PRINT MICROPGM MEMORY
000002  000276'                         .WORD   PR1MM
000004    115     104                   .ASCII  /MD/                    ;PRINT DATA MEMORY
000006  000476'                         .WORD   PR1MD
        000002                  PR1LN   ==      <.-PR1TBL>/4
                                ;
                                ;
                                ;       PRINT
                                ;       PERFORM THIRD LEVEL PARSING
                                ;       EG. IN THE COMMAND:
                                ;       MR>PR MD 0
                                ;       PARSE THE 'MD'
                                ;
                                ;
000010                          PR1::
000010                                  CALL    FIND                    ;LOCATE MEMORY MNEMONIC IN COMMAND LINE
000014  103004                         BCC     1$                      ;OK, CONTINUE
000016                                  CALL    ERR4                    ;'MISSING OPERAND'
000022  000167  001042                  JMP     PR1X                    ;EXIT
                                ;
000026  012700  000002          1$:     MOV     #PR1LN,R0               ;NUMBER OF TABLE ENTRIES
000032  012702  000000'                 MOV     #PR1TBL,R2              ;POINT TO TABLE
000036                                  CALL    SCAN                    ;MATCH AGAINST COMMAND LINE
000042  103004                         BCC     2$                      ;MATCH WAS MADE
000044                                  CALL    ERR7                    ;'INVALID MEMORY MNEMONIC'
```

```
 58 000050  000167  001014                  JMP·····PR1X
 59                                 ;
 60                                 ;       SAVE·POINTER·TO·ROUTINE·ASSOCIATED·WITH·THE·MEMORY·MNEMONIC.
 61                                 ;       SCAN·THE·REMAINDER·OF·THE·COMMAND·LINE·FOR·CONTROL·INFORMATION:
 62·                                ;
 63                                 ;       1. START·ADDRESS·ONLY. PRINT·ONE MEMORY·LOCATION.
 64                                 ;          MR>PR·MD·0
 65                                 ;       2. START·ADDRESS, LOOP INDICATOR. LOOP·ON·THE·READING·OF·
 66                                 ;          THIS·ONE·MEMORY·LOCATION·ONLY. PRINT·THE·CONTENTS ONLY·
 67                                 ;          ONCE.
 68                                 ;          MR>LD·MD·0 L·
 69                                 ;       3. START·ADDRESS, END·ADDRESS, NO·LOOP. PRINT·THE·CONTENTS·
 70                                 ;          OF·MEMORY BETWEEN·THE·START·AND·END·ADDRESSES.
 71                                 ;          MR>PR·MD·0 7
 72·                                ;       4. START·ADDRESS, END·ADDRESS, LOOP·INDICATOR. READ·THE·CONTENTS·
 73                                 ;          OF·MEMORY BETWEEN·THE·START·AND·END·ADDRESSES. HOWEVER, PRINT·
 74                                 ;          ONLY·THE·CONTENTS·OF·MEMORY·AT·THE·START·ADDRESS.
 75                                 ;          MR>PR·MD·0 7 L·
 76                                 ;
 77                                 ;
 78 000054  010167  000000G·        2$:     MOV·    R1,RTNPT·               ;SAVE·POINTER·
 79 000060                                  CALL·   FIND                    ;LOCATE·NON-BLANK·IN COMMAND·LINE·
 80 000064  103004                          BCC·    3$                      ;OK, CONTINUE·
 81 000066                                  CALL·   ERR4                    ;'MISSING·OPERAND'
 82·000072· 000167  000772                  JMP·    PR1X                    ;EXIT·
 83 000076                          3$:     CALL·   PACK                    ;CONVERT·COMMAND·LINE·VALUE·TO BINARY·
 84 000102· 103004                          BCC·    4$                      ;CONVERSION·SUCCESSFUL·
 85 000104                                  CALL·   ERR5                    ;'INVALID·NUMERIC·VALUE'
 86 000110  000167  000754                  JMP·    PR1X
 87                                 ;
 88 000114  016767  000000G·000000G· 4$:     MOV·    BINWD,MSTRT·            ;SAVE·PRINT·START·ADDRESS·
 89 000122· 016767  000000G·000000G·        MOV·    BINWD,MSTR2·            ;SAVE·IT·TWICE·
 90 000130  012767  177777  000000G·        MOV·    #-1,MEND·               ;INIT·END·ADDRESS·
 91                                 ;
 92·                                ;       START·ADDRESS·HAS·BEEN FOUND. SCAN·FOR·END ADDRESS·OR·
 93                                 ;       LOOP·INDICATOR·(CONDITIONS·2, 3 ABOVE). IF THERE·IS·NOTHING·
 94                                 ;       FURTHER·IN·THE·COMMAND LINE, CONDITION·1 IS·IN·EFFECT.
 95                                 ;
 96 000136                                  CALL·   FIND                    ;SCAN·COMMAND·LINE·
 97 000142· 103004                          BCC·    5$                      ;SOMETHING THERE·
 98 000144  016767  000000G·000000G·        MOV·    MSTRT,MEND·             ;SET·END·ADDR·=·START·ADDR·
 99 000152· 000445                          BR       9$                     ;JUMP·TO·RTN·
100                                 ;
101 000154  122711  000114          5$:     CMPB·   #'L,(R1)                ;LOOP·INDICATOR·
102·000160  001011                         BNE·    6$                      ;NO, MUST·BE·UPPER·ADDRESS·
103 000162· 052767  000000G·000000G·       BIS·    #OUT.BASE·              ;SET·FLAG·FOR·OUTPUT CONTROL·
104 000170  016767  000000G·000000G·       MOV·    MSTRT,MEND·             ;SET·END·ADDR·=·START·ADDR·
105 000176                                 CALL·   HANG                    ;HOW·TO·STOP·LOOP·
106 000202· 000431                         BR      9$                      ;JUMP·TO·RTN·
107                                ;
108 000204                         6$:     CALL·   PACK                    ;CONVERT·UPPER·ADDRESS·
109 000210  103004                         BCC·    7$                      ;OK, CONTINUE·
110 000212·                                CALL·   ERR5                    ;INVALID·NUMERIC·
111 000216  000167  000646                 JMP·    PR1X                    ;EXIT·
112·                               ;
113                                ;       SAVE·END·ADDRESS·(BINARY)
114                                ;       CHECK·FOR·LOOP·INDICATOR·AFTER·END·ADDRESS·(CONDITION·4)·
```

```
                                        ;
000222  016767  000000G 000000G 7$:     MOV     BINWD,MEND              ;SET UP END ADDRESS
000230                                  CALL    FIND                    ;CHECK FOR LOOP INDICATOR
000234  103414                         BCS     9$                      ;NO LOOP
000236  122711  000114                 CMPB    #'L,(R1)                ;CORRECT LOOP INDICATOR
000242  001404                         BEQ     8$                      ;YES
000244                                  CALL    ERR11
000250  000167  000614                 JMP     PR1X                    ;NO
000254  052767  000000G 000000G 8$:     BIS     #OUT,BASE               ;SET OUTPUT CONTROL
000262                                  CALL    HANG                    ;HOW TO STOP LOOP
                                        ;
000266  016701  000000G          9$:    MOV     RTNPT,R1                ;POINT TO ROUTINE
000272  000171  000000                   JMP     @(R1)                   ;JUMP TO ROUTINE
                                        ;
                                        ;
                                        ;       PRINT FROM MICROPGM MEMORY
                                        ;
000276                                  PR1MM::
000276  016746  000000G                 MOV     MMHIGH,-(SP)            ;SUPPLY UPPER MEMORY LIMIT
000302  016746  000000G                 MOV     MMLOW,-(SP)             ;LOWER LIMIT
000306                                  CALL    BUFS2                   ;PREPARE FOR LOAD
000312  103002                         BCC     1$                      ;NO ERROR
000314  000167  000550                 JMP     PR1X                    ;ERROR
                                        ;
000320  016746  000000G          1$:    MOV     MSTR2,-(SP)             ;SEND CURRENT ADDRESS
000324                                  CALL    SEQMM                   ;SEQUENCE UP TO PRINT START ADDRESS
000330  012746  000006                 MOV     #<MMOE+MMLEFT>,-(SP)
000334                                  CALL    MRPCR
000340  005046                         CLR     -(SP)                   ;DIRECT CNTL WORD TO MRP
000342                                  CALL    LBMRP                   ;ISSUE CLOCK TO REINHIBIT BR SELECT
000346                                  CALL    MRPLB                   ;REQUEST MRP TO LOD BUS         NS
000352  012667  000000G                 MOV     (SP)+,DATA1             ;GET MRP WORD FROM STACK
                                        ;
000356  016746  000000G                 MOV     MSTR2,-(SP)             ;SEND CURRENT ADDRESS
000362                                  CALL    SEQMM                   ;SEQUENCE UP TO PRINT START ADDRESS
000366  012746  000004                 MOV     #<MMOE>,-(SP)           ;SELECT RIGHT HALF OF PLR
000372                                  CALL    MRPCR                   ;DIRECT CNTL WORD TO MRP
000376  005046                         CLR     -(SP)
000400                                  CALL    LBMRP                   ;CLOCK TO REINHIBIT BRANCH REGISTER
000404                                  CALL    MRPLB                   ;REQUEST MRP TO LOD BUS         NS
000410  012667  000000G                 MOV     (SP)+,DATA2             ;GET MRP WORD FROM STACK
000414  005046                         CLR     -(SP)                   ;CLEAR MRP CONTROL REG
000416                                  CALL    MRPCR
                                        ;
000422                                  CALL    PRDATA                  ;PRINT MEMORY CONTENTS
000426  103002                         BCC     2$                      ;NOT END OF MEMORY
000430  000167  000434                 JMP     PR1X                    ;END OF MEMORY
                                        ;
000434  066767  000000G 000000G 2$:     ADD     INCVAL,MSTR2            ;ADVANCE ADDRESS
000442  026767  000000G 000000G        CMP     MSTR2,MEND              ;HAS UPPER MEMORY LIMIT BEEN REACHED
000450  101723                         BLOS    1$                      ;NO, CONTINUE
000452  032767  000000G 000000G        BIT     #LOOP,BASE              ;LOOP ON ?
000460  001002                         BNE     3$                      ;YES, CONTINUE
000462  000167  000402                 JMP     PR1X                    ;EXIT
000466  016767  000000G 000000G 3$:     MOV     MSTRT,MSTR2             ;INIT START ADDRESS
000474  000711                         BR      1$                      ;AND REPEAT PRINT
                                        ;
```

```
                                        ;
                                        ;       PRINT FROM DATA MEMORY
                                        ;
000476                                  PR1MD::
000476  016746  000000G                         MOV     MDHIGH,-(SP)            ;SUPPLY MEMORY UPPER LIMIT
000502  016746  000000G                         MOV     MDLOW,-(SP)             ;LOWER LIMIT
000506                                          CALL    BUFSET                  ;PREPARE FOR LOAD
000512  103566                                  BCS     PR1X                    ;ERROR
                                        ;
000514  012746  000377                          MOV     #377,-(SP)              ;SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
000520                                          CA__    SEQMM
                                        ;
000524  005046                                  CLR     -(SP)                   ;REINHIBIT BRANCH CONTROL REG
000526                                          CALL    MRPCR
000532  005046                                  CLR     -(SP)
000534                                          CALL    LBMRP                   ;SINGLE CLOCK TO REINHIBIT BRANCH REGISTER
                                        ;
000540  005046                                  CLR     -(SP)                   ;START MICROCODE AT 0
000542                                          CALL    SEQCS
                                        ;
000546  005046                                  CLR     -(SP)                   ;REINHIBIT BRANCH CONTROL REG
000550                                          CALL    CPCR
000554  005046                                  CLR     -(SP)
000556                                          CALL    LBCP                    ;SINGLE CLOCK TO REINHIBIT BRANCH REGISTER
                                        ;
000562  012767  001000  176422                  MOV     #Q$REBK,QR$CR2          ;RE-ARM INTERRUPTS
000570  012767  120000  176422                  MOV     #<Q$SM+Q$ENOP>,QR$CR2   ;SET SEARCH MODE + ENABLE INTERRUPTS
000576  012746  000360                          MOV     #Q$CSEL,-(SP)           ;CLEAR ALL SELECTIONS
000602  052716  001001                          BIS     #<Q$LBD+Q$LBP>,(SP)     ;CLEAR DRIVE AND PULSE
000606  052716  170000                          BIS     #<Q$MNC+Q$CNC>,(SP)     ;CLEAR CP NO-CLOCK
000612  005046                                  CLR     -(SP)                   ;SET NOTHING
000614                                          CALL    CSR1
                                        ;
000620  012767  000006  176424          PRMD:   MOV     #Q$RDMD,QR$LBR          ;MOVE ATTN CODE TO LOD BUS REG
000626  012767  120100  176422                  MOV     #<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2 ;SET ATTN CODE READY
000634  016701  176422                  1$:     MOV     QR$CR2,R1               ;READ CSR2
000640  032701  000100                          BIT     #Q$ATTN,R1              ;ATTN CLEAR
000644  001373                                  BNE     1$                      ;NO, READ AGAIN
                                        ;
000646  016767  000000G 176424                  MOV     MSTR2,QR$LBR            ;CD MEMORY START ADDRESS
000654  012767  120040  176422                  MOV     #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
000662  016701  176422                  2$:     MOV     QR$CR2,R1               ;READ CSR2
000666  032701  000040                          BIT     #Q$CCCP,R1              ;IS CC TO CP CLEAR
000672  001373                                  BNE     2$                      ;NO, READ AGAIN
                                        ;
000674  012767  000001  176424                  MOV     #1,QR$LBR               ;TRANSFER COUNT = 1 WORD
000702  012767  120040  176422                  MOV     #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
000710  016701  176422                  3$:     MOV     QR$CR2,R1               ;READ CSR2
000714  032701  000040                          BIT     #Q$CCCP,R1              ;IS CC TO CP CLEAR
000720  001373                                  BNE     3$                      ;NO, READ AGAIN
                                        ;
000722  012767  000000G 176424                  MOV     #DATA1,QR$LBR           ;CC MEMORY DATA BUFFER
000730  012767  120040  176422                  MOV     #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
                                        ;
                                        ;       WAIT FOR INTERRUPT FROM CP
                                        ;
000736                                          WT3E$S  #EFN,3
```

```
                                        ;
000750                                          CLEF$S  #EFN.3
                                        ;
                                        ;       RE-ARM INTERRUPTS
                                        ;
000762  012767  100400  176422                   MOV     #<Q$SM+Q$CHB>,QR$CR2    ;CLEAR INTERRUPT (USE HIT BUFFER INT)
000770  012767  101000  176422                   MOV     #<Q$SM+Q$REBK>,QR$CR2   ;RE-ARM
000776  012767  160000  176422                   MOV     #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2 ;ENABLE
                                        ;
001004                                          CALL    PRDATA                  ;PRINT MEMORY CONTENTS
001010  103417                                  BCS     PRMDX                   ;END OF MEMORY, EXIT
                                        ;
001012  066767  000000G 000000G                 ADD     INCVAL,MSTR2            ;ADVANCE ADDRESS
001020  026767  000000G 000000G                 CMP     MSTR2,MEND              ;HAS UPPER MEMORY LIMIT BEEN REACHED
001026  101674                                  BLOS    PRMD                    ;NO, CONTINUE
001030  032767  000000G 000000G                 BIT     #LOOP,BASE              ;LOOP ON ?
001036  001404                                  BEQ     PRMDX                   ;NO, EXIT
001040  016767  000000G 000000G                 MOV     MSTRT,MSTR2             ;INIT START ADDRESS
001046  000664                                  BR      PRMD                    ;AND REPEAT PRINT
                                        ;
001050  005046                          PRMDX:  CLR     -(SP)                   ;CLEAR NOTHING IN CSR1
001052  012746  176000                          MOV     #Q$NCLK,-(SP)           ;SET NO-CLOCKS
001056                                          CALL    CSR1
001062  012767  000000  176422                   MOV     #0,QR$CR2               ;SET LOAD MODE
                                        ;
                                        ;
001070                                  PR1X:
001070  042767  000000C 000000G                 BIC     #<ONCE+OUT>,BASE        ;CLEAR PRINT CONTROL FLAGS
001076                                          CALL    KILL                    ;KILL AST (IF THERE WAS ONE)
001102  000167  000000G                         JMP     MRPXX
                                        ;
                    000001                      .END
```

```
ALUCKE= 040000          BYTE39= 000047          BYTE90= 000132          MDHIGH= ****** GX       Q$FC   = 000045
ALUOE = 004000          BYTE4 = 000004          BYTE91= 000133          MDLOW = ****** GX       Q$FO   = 000044
A01   = 010000          BYTE40= 000050          BYTE92= 000134          MEND  = ****** GX       Q$FP   = 000046
BASE  = ****** GX       BYTE41= 000051          BYTE93= 000135          MMADRD= 000100          Q$HBF  = 000002
BINWD = ****** GX       BYTE42= 000052          BYTE94= 000136          MMHIGH= ****** GX       Q$ICP  = 000006
BITVAL= 000000          BYTE43= 000053          BYTE95= 000137          MMLEFT= 000002          Q$IHB  = 000003
BIT0  = 000001          BYTE44= 000054          BYTE96= 000140          MMLOW = ****** GX       Q$IHRL= 000002
BIT1  = 000002          BYTE45= 000055          BYTE97= 000141          MMOE  = 000004          Q$IMRP= 000007
BIT10 = 002000          BYTE46= 000056          BYTE98= 000142          MMWRTE= 000010          Q$LBD  = 001000
BIT11 = 004000          BYTE47= 000057          BYTE99= 000143          MNOBRE= 100000          Q$LBDP= 001001
BIT12 = 010000          BYTE48= 000060          BYTVAL= 000144          MREN1 = 000001          Q$LBP  = 000001
BIT13 = 020000          BYTE49= 000061          CBKALL= 001000          MREN2 = 020000          Q$LDCD= 000003
BIT14 = 040000          BYTE5 = 000005          CBKCLK= 000400          MRPCR = ****** GX       Q$LDMD= 000004
BIT15 = 100000          BYTE50= 000062          CNOBRE= 100000          MRPLB = ****** GX       Q$LDPP= 002000
BIT2  = 000004          BYTE51= 000063          CPCCEN= 010000          MRPXX = ****** GX       Q$LHP  = 010000
BIT3  = 000010          BYTE52= 000064          CPCR  = ****** GX       MSTRT = ****** GX       Q$MNC  = 140000
BIT4  = 000020          BYTE53= 000065          CPREAD= 040000          MSTR2 = ****** GX       Q$MR   = 000052
BIT5  = 000040          BYTE54= 000066          CPWRTE= 020000          MSYN  = 000040          Q$MRP  = 000040
BIT6  = 000100          BYTE55= 000067          CSADRD= 000004          N     = 000144          Q$MRP2= 000240
BIT7  = 000200          BYTE56= 000070          CSEQCI= 100000          ONCE  = ****** GX       Q$MSC  = 040000
BIT8  = 000400          BYTE57= 000071          CSOE  = 000040          OUT   = ****** GX       Q$MSET= 000004
BIT9  = 001000          BYTE58= 000072          CSR1  = ****** GX       PACK  = ****** GX       Q$MSP  = 100000
BUFSET= ****** GX       BYTE59= 000073          CSWRTE= 000100          PLB   = 000010          Q$NCLK= 176000
BUFS2 = ****** GX       BYTE6 = 000006          DATA1 = ****** GX       PLC   = 000020          Q$PP   = 000100
BYTE0 = 000000          BYTE60= 000074          DATA2 = ****** GX       PLD   = 000030          Q$PPSW= 000320
BYTE1 = 000001          BYTE61= 000075          DBR.RD= 000001          PLRWR = 000200          Q$PP2  = 000300
BYTE10= 000012          BYTE62= 000076          DB$CPP= 001457          PLR.EN= 000200          Q$QHLT= 000013
BYTE11= 000013          BYTE63= 000077          DB$SPT= 000026          PRDATA= ****** GX       Q$QL   = 000043
BYTE12= 000014          BYTE64= 000100          DB$TPC= 000023          PRMD    000620R         002 Q$QLA = 000053
BYTE13= 000015          BYTE65= 000101          DISPGS= 100000          PRMDX   001050R         002 Q$QLB = 000054
BYTE14= 000016          BYTE66= 000102          DMAAWR= 000005          PR1     000010RG        002 Q$QLR = 000001
BYTE15= 000017          BYTE67= 000103          DMARRD= 000003          PRILN = 000002 G            Q$QW   = 000042
BYTE16= 000020          BYTE68= 000104          DMARWR= 000004          PRIMD   000476RG        002 Q$RDCD= 000005
BYTE17= 000021          BYTE69= 000105          EFN.3 = ****** GX       PRIMM   000276RG        002 Q$RDMD= 000006
BYTE18= 000022          BYTE7 = 000007          ENBR  = 010000          PRITBL  000000RG        002 Q$REBK= 001000
BYTE19= 000023          BYTE70= 000106          ERR11 = ****** GX       PRIX    001070R         002 Q$RNC = 006000
BYTE2 = 000002          BYTE71= 000107          ERR4  = ****** GX       QR$CR1= 176420              Q$RSC  = 004000
BYTE20= 000024          BYTE72= 000110          ERR5  = ****** GX       QR$CR2= 176422              Q$RSET= 000010
BYTE21= 000025          BYTE73= 000111          ERR7  = ****** GX       QR$LBR= 176424              Q$SM   = 100000
BYTE22= 000026          BYTE74= 000112          FIND  = ****** GX       Q$ATTN= 000100              Q$SP   = 000120
BYTE23= 000027          BYTE75= 000113          HANG  = ****** GX       Q$BCL = 000001              Q$SP2 = 000340
BYTE24= 000030          BYTE76= 000114          INCVAL= ****** GX       Q$CCCP= 000040              RGO.EN= 000200
BYTE25= 000031          BYTE77= 000115          KILL  = ****** GX       Q$CHB = 000400              RGO.VA= 020000
BYTE26= 000032          BYTE78= 000116          LBCP  = ****** GX       Q$CHRL= 000200              RTNPT = ****** GX
BYTE27= 000033          BYTE79= 000117          LBMRP = ****** GX       Q$CLR = 000040              SCAN  = ****** GX
BYTE28= 000034          BYTE8 = 000010          LOC.EN= 000100          Q$CNC = 030000              SEQCS = ****** GX
BYTE29= 000035          BYTE80= 000120          LOC.WA= 040000          Q$CP  = 000060              SEQMM = ****** GX
BYTE3 = 000003          BYTE81= 000121          LOC.WB= 100000          Q$CPCC= 000010              SEQ.CI= 000010
BYTE30= 000036          BYTE82= 000122          LOOP  = ****** GX       Q$CP2 = 000260              S$CLR = 000000
BYTE31= 000037          BYTE83= 000123          MAREN1= 000001          Q$CSC = 010000              S$LA   = 000001
BYTE32= 000040          BYTE84= 000124          MAREN2= 004000          Q$CSEL= 000360              S$OB   = 000005
BYTE33= 000041          BYTE85= 000125          MARLOD= 010000          Q$CSET= 000002              S$OR   = 000006
BYTE34= 000042          BYTE86= 000126          MAROUT= 000002          Q$CSP = 020000              S$OX   = 000004
BYTE35= 000043          BYTE87= 000127          MAR.LO= 002000          Q$DMA = 000001              S$SR   = 000007
BYTE36= 000044          BYTE88= 000130          MAR.OU= 000040          Q$ENBK= 040000              S$S1   = 000010
BYTE37= 000045          BYTE89= 000131          MBKALL= 001000          Q$ENOP= 020000              S$S2   = 000014
BYTE38= 000046          BYTE9 = 000011          MBKCLK= 000400          Q$FAL = 004000              TD$CTR= 176370
```

```
TD$CTW= 176360          T$ICD = 000040          WORD18= 000044          WORD46= 000134          WORD74= 000224
TD$INL= 004000          T$MODE= 004000          WORD19= 000046          WORD47= 000136          WORD75= 000226
TD$MEM= 000270          T$OB  = 000036          WORD2 = 000004          WORD48= 000140          WORD76= 000230
TD$OAR= 176344          T$OBE = 004000          WORD20= 000050          WORD49= 000142          WORD77= 000232
TD$OTR= 176346          T$OBF = 010000          WORD21= 000052          WORD5 = 000012          WORD78= 000234
TD$QRD= 000274          T$OBRA= 000034          WORD22= 000054          WORD50= 000144          WORD79= 000236
TD$SW = 176376          T$OBWA= 000032          WORD23= 000056          WORD51= 000146          WORD8 = 000020
TD$TAR= 176372          T$OUTA= 100000          WORD24= 000060          WORD52= 000150          WORD80= 000240
TD$TAW= 176362          T$RBDO= 000200          WORD25= 000062          WORD53= 000152          WORD81= 000242
TD$TDR= 176374          T$RNB = 000040          WORD26= 000064          WORD54= 000154          WORD82= 000244
TD$TDW= 176364          T$RSET= 040000          WORD27= 000066          WORD55= 000156          WORD83= 000246
T$AD  = 000020          T$SC  = 000022          WORD28= 000070          WORD56= 000160          WORD84= 000250
T$BA  = 000002          T$SCLK= 020000          WORD29= 000072          WORD57= 000162          WORD85= 000252
T$BD  = 000010          T$SEG1= 000000          WORD3 = 000006          WORD58= 000164          WORD86= 000254
T$BSO = 100000          T$SEG2= 000001          WORD30= 000074          WORD59= 000166          WORD87= 000256
T$BT  = 000020          T$SEG3= 000002          WORD31= 000076          WORD6 = 000014          WORD88= 000260
T$BTAR= 000030          T$SO  = 000001          WORD32= 000100          WORD60= 000170          WORD89= 000262
T$BTD = 002000          T$UBUS= 100000          WORD33= 000102          WORD61= 000172          WORD9 = 000022
T$CD  = 000100          T$1CLK= 000400          WORD34= 000104          WORD62= 000174          WORD90= 000264
T$CLK = 002000          T$8BEN= 000020          WORD35= 000106          WORD63= 000176          WORD91= 000266
T$DISK= 000200          UBD.IN= 000020          WORD36= 000110          WORD64= 000200          WORD92= 000270
T$DRD = 000004          WORD0 = 000000          WORD37= 000112          WORD65= 000202          WORD93= 000272
T$EMEM= 010000          WORD1 = 000002          WORD38= 000114          WORD66= 000204          WORD94= 000274
T$FSAA= 000000          WORD10= 000024          WORD39= 000116          WORD67= 000206          WORD95= 000276
T$FSAB= 000004          WORD11= 000026          WORD4 = 000010          WORD68= 000210          WORD96= 000300
T$FSAC= 000014          WORD12= 000030          WORD40= 000120          WORD69= 000212          WORD97= 000302
T$FSB2= 000010          WORD13= 000032          WORD41= 000122          WORD7 = 000016          WORD98= 000304
T$IB  = 000026          WORD14= 000034          WORD42= 000124          WORD70= 000214          WORD99= 000306
T$IBAR= 000024          WORD15= 000036          WORD43= 000126          WORD71= 000216          WRDVAL= 000310
T$IBE = 020000          WORD16= 000040          WORD44= 000130          WORD72= 000220          XTREAD= 001000
T$IBF = 040000          WORD17= 000042          WORD45= 000132          WORD73= 000222          XTWRTE= 000400

. ABS.  000000     000
        000000     001
MRPR    001106     002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3569 WORDS  ( 14 PAGES)
DYNAMIC MEMORY:  4916 WORDS  ( 18 PAGES)
ELAPSED TIME:  00:00:47
MRPR,MRPR/-SP=[20,1]IM,[20,1]MRPR
```

```
                                        .TITLE  MRREST
000000                                  .PSECT  MRREST
                                ;
                                ;
                                ;       HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
                                ;       MATCH REPORT PROCESSOR ROUTINES
                                ;
                                ;       COMMANDS:
                                ;       ST      STORE INTO AN MRP REGISTER
                                ;       RE      READ FROM A REGISTER
                                ;       RS      RESET MRP
                                ;       CL      CALL HQR LOADER
                                ;
                                ;       ONCE A COMMAND HAS BEEN EXECUTED (OR AN ERROR ENCOUNTERED)
                                ;       THIS MODULE RETURNS CONTROL TO THE MODULE MRP AT LOCATION
                                ;       'MRPXX'.
                                ;
                                ;
                                ;       STORE
                                ;       TABLE OF VALID REGISTER MNEMOONICS AND ASSOCIATED ROUTINE
                                ;       ADDRESSES
                                ;
000000                          ST1TBL::
000000  103     122                     .ASCII  /CR/                    ;CONTROL REG
000002  000216'                         .WORD   ST1CR
000004  102     113                     .ASCII  /BK/                    ;BREAKPOINT REG
000006  000230'                         .WORD   ST1BK
000010  115     101                     .ASCII  /MA/                    ;MEMORY ADDRESS REG
000012  000252'                         .WORD   ST1MA
000014  120     114                     .ASCII  /PL/                    ;PIPELINE REG LEFT
000016  000314'                         .WORD   ST1PL
000020  120     122                     .ASCII  /PR/                    ;PIPELINE REG RIGHT
000022  000336'                         .WORD   ST1PR
000024  121     122                     .ASCII  /QR/                    ;Q-REG
000026  000360'                         .WORD   ST1QR
000030  114     127                     .ASCII  /LW/                    ;'WORD LOCATION'
000032  000432'                         .WORD   ST1LW
000034  114     123                     .ASCII  /LS/                    ;'SENTENCE' LOCATION
000036  000440'                         .WORD   ST1LS
000040  114     120                     .ASCII  /LP/                    ;'PARAGRAPH' LOCATION
000042  000446'                         .WORD   ST1LPG
        000011                  ST1LN   ==      <.-ST1TBL>/4
                                ;
                                ;       LOCATION WORD WORD AREAS
                                ;
000044  000000                  LOCWRD: .WORD   0
000046  000000  040000  100000  LOCTBL: .WORD   0,BIT14,BIT15,BIT15+BIT14
000054  140000
                                ;
                                ;
                                ;
                                ;       READ
                                ;       TABLE OF VALID REGISTER MNEMONICS AND ASSOCIATED ROUTINE
                                ;       ADDRESSES
                                ;
000056                          RE1TBL::
000056  115     101                     .ASCII  /MA/                    ;READ MAR
```

```
 57 000060  000564'                     .WORD   RE1MA
 58 000062     115      120             .ASCII  /MP/                    ;READ MICROPGM ADDR
 59 000064  000612'                     .WORD   RE1MP
 60 000066     123      127             .ASCII  /SW/                    ;READ UNIBUS STATUS WORD
 61 000070  000644'                     .WORD   RE1SW
 62 000072     104      127             .ASCII  /DW/                    ;READ UNIBUS DATA WORD
 63 000074  000736'                     .WORD   RE1DW
 64         000004             RE1LN:  ==      <.-RE1TBL>/4
 65                                    ;
 66                                    ;
 67                                    ;       STORE
 68                                    ;       PERFORM THIRD LEVEL PARSING
 69                                    ;       EG. IN THE COMMAND:
 70                                    ;       MR>ST 0 CR
 71                                    ;       PARSE THE '0'
 72                                    ;
 73                                    ;
 74 000076                             ST1::
 75 000076                                     CALL    FIND                    ;FIND OPERAND IN COMMAND LINE
 76 000102  103004                             BCC     1$                      ;OK, CONTINUE
 77 000104                                     CALL    ERR4                    ;NOTHING THERE
 78 000110  000167  000354                     JMP     ST1X                    ;RETURN TO MRP (PROMPT)
 79 000114                             1$:     CALL    PACK                    ;CONVERT VALUE IN COMMAND LINE TO BINARY
 80 000120  103004                             BCC     2$
 81 000122                                     CALL    ERR5                    ;INVALID NUMERIC
 82 000126  000167  000336                     JMP     ST1X
 83                                    ;
 84                                    ;       CONTINUE PARSING THE COMMAND LINE
 85                                    ;       FIND THE REGISTER MNEMONIC
 86                                    ;
 87 000132                             2$:     CALL    FIND                    ;LOCATE A NON-BLANK
 88 000136  103004                             BCC     3$
 89 000140                                     CALL    ERR4                    ;NOTHING THERE - ERROR
 90 000144  000167  000320                     JMP     ST1X
 91                                    ;
 92                                    ;       MATCH REGISTER MNEMONIC FROM THE COMMAND LINE AGAINST
 93                                    ;       THE TABLE OF VALID MNEMONICS.
 94                                    ;
 95 000150  012700  000011             3$:     MOV     #ST1LN,R0               ;NUMBER OF TABLE ENTRIES
 96 000154  012702  000000'                    MOV     #ST1TBL,R2              ;R2 -> TABLE
 97 000160                                     CALL    SCAN                    ;MATCH AGAINST COMMAND LINE
 98 000164  103004                             BCC     4$                      ;OK, CONTINUE
 99 000166                                     CALL    ERR6
100 000172  000167  000272                     JMP     ST1X
101                                    ;
102                                    ;       SAVE THE POINTER TO THE ROUTINE ASSOCIATED WITH THE
103                                    ;       REGISTER, R1 -> ROUTINE ADDRESS.
104                                    ;       CALL ROUTINE TO SCAN COMMAND LINE FOR LOOP INDICATOR.
105                                    ;       EG. MR>ST 0 CR L
106                                    ;       LOOP FLAG WILL BE SET IF INDICATOR IS PRESENT.
107                                    ;       JUMP TO ROUTINE TO LOAD REGISTER.
108                                    ;
109 000176  010167  000000G            4$:     MOV     R1,RTNPT                ;SAVE POINTER TO RTN
110 000202                                     CALL    LOOPR                   ;LOOP?
111 000206  016701  000000G            ST1IN:  MOV     RTNPT,R1                ;POINT TO ROUTINE
112 000212  000171  000000                     JMP     @(R1)                   ;EXECUTE ROUTINE
113                                    ;
```

```
                                        ;
                                        ;       MRP CONTROL REG
                                        ;
000216                                  ST1GR::
000216  016746  000000G                         MOV     BINWD,-(SP)             ;MOVE DATA TO LOD BUS REG
000222                                          CALL    MRPCR                   ;DIRECT CNTL WORD TO MRP
000226  000512                                  BR      ST1LP                   ;TEST LOOP FLAG
                                        ;
                                        ;       BREAKPOINT REG
                                        ;
000230                                  ST1BK::
000230  012746  000400                          MOV     #MBKCLKEN,-(SP)         ;SET MRP CNTL FOR BREAKPOINT
000234                                          CALL    MRPCR                   ;DIRECT CNTL WORD TO MRP
000240  016746  000000G                         MOV     BINWD,-(SP)             ;LOAD DATA WORD INTO LOAD BUS REG
000244                                          CALL    LBMRP                   ;SET BCE CNTL WORD
000250  000501                                  BR      ST1LP                   ;TEST LOOP FLAG
                                        ;
                                        ;       MEMORY ADDRESS REG
                                        ;
000252                                  ST1MA::
000252  012746  000200                          MOV     #<PLR.EN>,-(SP)         ;SET CNTL BITS FOR MRP
000256                                          CALL    MRPCR                   ;DIRECT CNTL WORD TO MRP
000262  012746  002000                          MOV     #<MAR.LOD>,-(SP)        ;CNTL BITS TO LOAD MAR          NS
000266                                          CALL    LBMRP                   ;SEND THEM TO MRP
000272  012746  000001                          MOV     #MAREN1, -(SP)          ;CLEAR PLR-RT ENABLE BITS       NS
000276                                          CALL    MRPCR                   ;AND SET MAREN1 IN CR           NS
000302  016746  000000G                         MOV     BINWD,-(SP)             ;MOVE DATA WORD TO LOD BUS REG
000306                                          CALL    LBMRP                   ;SEND DATA TO MRP (MAR)
000312  000460                                  BR      ST1LP
                                        ;
                                        ;       PIPELINE REG LEFT
                                        ;
000314                                  ST1PL::
000314  012746  000202                          MOV     #<PLR.EN+MMLEFT>,-(SP)  ;BIT PATTERN FOR PLR LEFT
000320                                          CALL    MRPCR                   ;DIRECT CNTL WORD TO MRP
000324  016746  000000G                         MOV     BINWD,-(SP)             ;MOVE DATA WORD TO LOD BUS REG
000330                                          CALL    LBMRP                   ;SEND DATA WWORD TO MRP
000334  000447                                  BR      ST1LP
                                        ;
                                        ;       PIPELINE REG RIGHT
                                        ;
000336                                  ST1PR::
000336  012746  000200                          MOV     #<PLR.EN>,-(SP)         ;CNTL WORD FOR PLR RT
000342                                          CALL    MRPCR                   ;DIRECT CNTL WORD TO MRP
000346  016746  000000G                         MOV     BINWD,-(SP)             ;MOVE DATA WORD TO LOD BUS REG
000352                                          CALL    LBMRP                   ;SEND DATA WORD TO MRP
000356  000436                                  BR      ST1LP
                                        ;
                                        ;       Q-REG
                                        ;
000360                                  ST1QR::
000360  012746  000200                          MOV     #<PLR.EN>,-(SP)         ;BIT PATTN FOR PLR RT
000364                                          CALL    MRPCR                   ;DIRECT CNTL WORD TO MRP
000370  012746  000200                          MOV     #RGQ.EN,-(SP)           ;ENABLE Q REG LOADING FROM MRP
000374                                          CALL    LBMRP                   ;SEND PATTN TO MRP
                                        ;
000400  012746  000202                          MOV     #<PLR.EN+MMLEFT>,-(SP)  ;BIT PATTERN FOR PLR LEFT
```

```
171 000404                                      CALL·   MRPCR·                  ;DIRECT·CNTL·WORD·TO·MRP·
172·000410  012746  020000·                     MOV·    #RGO.VAL,-(SP)          ;BIT·PATTN·TO·GENERATE·Q-REG·RDY·
173 000414                                      CALL·   LBMRP·                  ;SEND·PATTN·TO·MRP·
174 000420  016746  000000G·                    MOV·    BINWD,-(SP)             ;DATA·FOR·DESTINATION·
175 000424                                      CALL·   LBMRP·                  ;SEND·TO·MRP·
176 000430  000411                              BR      ST1LP·
177                                     ;
178                                     ;       WORD·LOCATION·
179                                     ;
180 000432·                             ST1LW::
181 000432·                                     CALL·   LOC1W·
182·000436  000406                              BR      ST1LP·
183                                     ;
184                                     ;       SENTENCE·LOCATION·
185                                     ;
186 000440                              ST1LS::
187 000440                                      CALL·   LOC1S·
188 000444  000403                              BR      ST1LP·
189                                     ;
190                                     ;       PARAGRAPH·LOCATION·
191                                     ;
192·000446                              ST1LPG::
193 000446                                      CALL·   LOC1P·
194 000452· 000400                              BR      ST1LP·
195                                     ;
196                                     ;
197 000454                              ST1LP:
198 000454  032767  000000G·000000G·            BIT·    #LOOP,BASE·             ;LOOP·ON·COMMAND·
199 000462· 001402·                             BEQ·    ST1X                    ;NO, EXIT·
200 000464  000167  177516                      JMP·    ST1IN·                  ;EXECUTE·ROUTINE·AGAIN·
201                                     ;
202·000470                              ST1X:
203 000470                                      CALL·   KILL                    ;KILL·AST·(IF·THERE·WAS·ONE)
204 000474  000167  000000G·                    JMP·    MRPXX·
```

```
                                        ;
                                        ;
                                        ;       READ
                                        ;       PERFORM THIRD LEVEL PARSING
                                        ;       EG. IN THE COMMAND:
                                        ;       MR>RE CR
                                        ;       PARSE THE 'CR'
                                        ;
                                        ;
000500                                  RE1::
000500                                          CALL    FIND                    ;FIND A REGISTER MNEMONIC IN COMMAND LINE
000504  103004                                  BCC     1$                      ;OK, CONTINUE
000506                                          CALL    ERR4                    ;NOTHING THERE
000512  000167  000376                          JMP     RE1X                    ;RETURN TO MRP (PROMPT)
                                        ;
                                        ;       MATCH THE REGISTER MNEMONIC FROM THE COMMAND LINE AGAINST
                                        ;       THE TABLE OF VALID MNEMONICS.
                                        ;
000516  012700  000004          1$:     MOV     #RE1LN,R0               ;NUMBER OF TABLE ENTRIES
000522  012702  000056'                 MOV     #RE1TBL,R2              ;R2 -> TABLE
000526                                  CALL    SCAN                    ;MATCH AGAINST COMMAND LINE
000532  103004                          BCC     2$                      ;OK, CONTINUE
000534                                  CALL    ERR6
000540  000167  000350                  JMP     RE1X
                                        ;
                                        ;       SAVE THE POINTER TO THE ROUTINE ASSOCIATED WITH THE
                                        ;       REGISTER, R1 -> ROUTINE ADDRESS.
                                        ;       CALL ROUTINE TO SCAN THE COMMAND LINE FOR LOOP INDICATOR.
                                        ;       EG. MR>RE CR L
                                        ;       LOOP FLAG WILL BE SET IF INDICATOR IS PRESENT.
                                        ;       JUMP TO ROUTINE TO READ REGISTER.
                                        ;
000544  010167  000000G         2$:     MOV     R1,RTNPT                ;SAVE POINTER TO RTN
000550                                  CALL    LOOPR                   ;LOOP?
000554  016701  000000G         RE1IN:  MOV     RTNPT,R1                ;POINT TO ROUTINE
000560  000171  000000                  JMP     @(R1)                   ;EXECUTE ROUTINE
                                        ;
                                        ;
                                        ;       MEMORY ADDRESS REG
                                        ;
000564                                  RE1MA::
000564  012746  000040                          MOV     #MAR.OUT,-(SP)
000570                                          CALL    MRPCR                   ;DIRECT CNTL WORD TO MRP
000574                                          CALL    MRPLB                   ;GET WORD FROM MRP
000600  005046                                  CLR     -(SP)                   ; CLEAR THE CONTROL REG
000602                                          CALL    MRPCR
000606  012601                                  MOV     (SP)+,R1                ;WORD RETURNED ON STACK
000610  000516                                  BR      RE1PUT
                                        ;
                                        ;       MICROPGM ADDRESS REG
                                        ;
000612                                  RE1MP::
000612  012746  000100                          MOV     #MMADRD,-(SP)
000616                                          CALL    MRPCR                   ;DIRECT CNTL WORD TO MRP
000622                                          CALL    MRPLB                   ;GET WORD FROM MRP
000626  005046                                  CLR     -(SP)                   ; CLEAR THE CONTROL REG
000630                                          CALL    MRPCR
```

```
 263 000634  012601                          MOV     (SP)+,R1                ;WORD RETURNED ON STACK
 264 000636  042701  177400                 BIC     #177400,R1              ;MASK OFF UNNECESSARY BITS
 265 000642  000501                          BR      RE1PUT
 266                                ;
 267                                ;       UNIBUS STATUS REG
 268                                ;
 269 000644                         RE1SW::
 270 000644  012746  000202                 MOV     #<PLR!EN+MMLEFT>,-(SP)  ;PL LEFT                NS
 271 000650                                 CALL    MRPCR                   ;DIRECT CNTL WORD TO MRP  NS
 272 000654  005046                         CLR     -(SP)                   ;DATA WORD              NS
 273 000656                                 CALL    LBMRP                   ;SEND WORD TO MRP
 274 000662  012746  000200                 MOV     #<PLR!EN>,-(SP)         ;PL RIGHT                NS
 275 000666                                 CALL    MRPCR                   ;DIRECT CNTL WORD TO MRP
 276 000672  012746  000060                 MOV     #<UBD!IN+MSYN>,-(SP)    ;MASTER SYNC
 277 000676                                 CALL    LBMRP                   ;SEND WORD TO MRP
 278 000702                                 CALL    MRPLB                   ;READ STATUS WORD FROM MRP
 279 000706  012746  000200                 MOV     #<PLR!EN>,-(SP)         ;CLEAR PLR.EN BIT       NS
 280 000712                                 CALL    MRPCR
 281 000716  005046                         CLR     -(SP)                   ;CLEAR PLR-RT                NS
 282 000720                                 CALL    LBMRP                   ;SEND WORD TO MRP            NS
 283 000724  005046                         CLR     -(SP)                   ;CLEAR THE CR                NS
 284 000726                                 CALL    MRPCR
 285 000732  012601                         MOV     (SP)+,R1                ;WORD RETURNED ON STACK
 286 000734  000444                         BR      RE1PUT                  ;AND PRINT
 287                                ;
 288                                ;       UNIBUS DATA WORD
 289                                ;
 290 000736                         RE1DW::
 291 000736  012746  000202                 MOV     #<PLR!EN+MMLEFT>,-(SP)  ;SET CR FOR PL LEFT         NS
 292 000742                                 CALL    MRPCR                   ;DIRECT CNTL WORD TO MRP     NS
 293 000746  012746  010000                 MOV     #A01, -(SP)             ; SET DATA WORD ADDRESS BIT NS
 294 000752                                 CALL    LBMRP                   ;SEND WORD TO MRP PLR-LEFT  NS
 295 000756  012746  000200                 MOV     #<PLR!EN>,-(SP)         ;SET CR FOR PL RIGHT    NS
 296 000762                                 CALL    MRPCR                   ;DIRECT CNTL WORD TO MRP
 297 000766  012746  000060                 MOV     #<UBD!IN+MSYN>,-(SP)    ;SET MASTER SYNC,UBD.IN IN PLR RT       NS
 298 000772                                 CALL    LBMRP                   ;SEND WORD TO MRP
 299 000776                                 CALL    MRPLB                   ;READ DATA WORD FROM MRP        NS
 300 001002  012746  000200                 MOV     #<PLR!EN>,-(SP)         ;CLEAR PLR.EN BIT       NS
 301 001006                                 CALL    MRPCR
 302 001012  005046                         CLR     -(SP)                   ;CLEAR PLR-RT
 303 001014                                 CALL    LBMRP                   ;SEND WORD TO MRP            NS
 304 001020  012746  000202                 MOV     #<MMLEFT+PLR!EN>,-(SP)  ;PL LEFT                 NS
 305 001024                                 CALL    MRPCR                   ;DIRECT CNTL WORD TO MRP          NS
 306 001030  005046                         CLR     -(SP)                   ;CLEAR PLR-LT
 307 001032                                 CALL    LBMRP                   ;CLEAR PLR LEFT             NS
 308 001036  005046                         CLR     -(SP)                   ;CLEAR MRP CR
 309 001040                                 CALL    MRPCR
 310 001044  012601                         MOV     (SP)+,R1                ;WORD RETURNED ON STACK
 311                                ;
 312                                ;
 313 001046  032767  000000G 000000G RE1PUT: BIT     #ONCE,BASE              ;PRINTED ONCE ?
 314 001054  001011                         BNE     1$                      ;YES, SKIP
 315 001056  052767  000000G 000000G        BIS     #ONCE,BASE              ;SET FLAG FOR PRINTED ONCE
 316 001064  012705  000000G                MOV     #PRINT,R5               ;POINT TO PRINT LINE
 317 001070                                 CALL    UNPK                    ;CONVERT VALUE IN R1 FOR PRINTING
 318 001074                                 CALL    CONSOL                  ;PRINT ON CONSOLE
 319                                ;
```

```
 320 001100  032767  000000G 000000G 1$:      BIT     #LOOP,BASE      ;REPEAT
 321 001106  001402                          BEQ     RE1X            ;NO, EXIT
 322 001110  000167  177440                  JMP     RE1IN           ;EXECUTE ROUTINE AGAIN
 323                                 ;
 324 001114                          RE1X:
 325 001114  042767  000000G 000000G        BIC     #ONCE,BASE      ;CLEAR PRINT CONTROL FLAG
 326 001122                                  CALL    KILL            ;KILL AST
 327 001126  000167  000000G                 JMP     MRPXX
```

```
                                        ;
                                        ;
                                        ;       RESET·
                                        ;
                                        ;
001132·                                 RS1::
001132·  005046                                 CLR·    -(SP)                   ;CLEAR·NOTHING·                 FW·
001134   012746  000004·                         MOV·    #0$MSET,-(SP)           ;SET·MRP·RESET·                 FW·
001140                                          CALL·   CSR1                    ;RESETS·MRP·                    FW·
001144   012746  000004                         MOV·    #0$MSET,-(SP)           ;CLEAR·RESET·                   FW·
001150   005046                                 CLR·    -(SP)                   ;SET·NOTHING·                   FW·
001152·                                         CALL·   CSR1                    ;ELIMINATES·MRP·RESET·          FW·
001156   000167  000000G·                       JMP·    MRPXX·
                                        ;
                                        ;
                                        ;
                                        ;       CALL·HQR·LOADER·
                                        ;
                                        ;
001162·                                 CL1::
001162·                                         CALL·   CL·                     ;CALL·ROUTINE·IN·'QMAIN'
001166   000167  000000G·                       JMP·    MRPXX·
```

```
                                        ;
                                        ;
                                        ;       STORE·INTO·LOCATION·WORDS·
                                        ;
                                        ;
001172· 005267  176646          LOC1P:: INC·    LOCWRD·                 ;PARAGRAPH
001176  005267  176642          LOC1S:: INC·    LOCWRD·                 ;SENTENCE·
001202·                         LOC1W::                                 ;WORD·
001202· 012746  000200                  MOV·    #<PLR.EN>,-(SP)         ;SET·BIT·PATTN·FOR·PLR·RT
001206                                  CALL·   MRPCR·                  ;DIRECT·CNTL·WORD·TO·MRP·
001212· 012746  000100                  MOV·    #LOC.EN,-(SP)           ;SET·BIT·PATTN·FOR·LOC·ENABLE·
001216                                  CALL·   LBMRP·                  ;SEND·WORD·TO·MRP·
                                        ;
001222· 012746  000202                  MOV·    #<PLR.EN+MMLEFT>,-(SP)  ;SET·BIT·PATTN·FOR·PLR·LEFT
001226                                  CALL·   MRPCR·                  ;DIRECT·CNTL·WORD·TO·MRP·
001232· 016700  176606                  MOV·    LOCWRD,R0               ;* DETERMINE·WHICH·LOCATION·
001236  006300                          ASL·    R0                      ;* WORD·TO·SEND·TO·
001240  016046  000046'                 MOV·    LOCTBL(R0),-(SP)        ;* THE·MRP
001244                                  CALL·   LBMRP·                  ;SEND·LOC·WORD·
                                        ;
001250  005046                          CLR·    -(SP)                   ; DISABLE·PIPELIN·REGISTER·  F.W. 8/31
001252·                                 CALL·   MRPCR·
001256  016746  000000G·                MOV·    BINWD,-(SP)             ;MOVE·USER·VALUE·TO·LOD·BUS·
001262·                                 CALL·   LBMRP·                  ;SEND·VALUE·TO·MRP·
001266  005067  176552                  CLR·    LOCWRD·                 ;RESET·
001272·                                 RETURN·
                                        ;
        000001                          .END·
```

```
ALUCKE= 040000
ALUOE = 004000
A01   = 010000
BASE  = ****** GX
BINWD = ****** GX
BITVAL= 000000
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BYTE0 = 000000
BYTE1 = 000001
BYTE10= 000012
BYTE11= 000013
BYTE12= 000014
BYTE13= 000015
BYTE14= 000016
BYTE15= 000017
BYTE16= 000020
BYTE17= 000021
BYTE18= 000022
BYTE19= 000023
BYTE2 = 000002
BYTE20= 000024
BYTE21= 000025
BYTE22= 000026
BYTE23= 000027
BYTE24= 000030
BYTE25= 000031
BYTE26= 000032
BYTE27= 000033
BYTE28= 000034
BYTE29= 000035
BYTE3 = 000003
BYTE30= 000036
BYTE31= 000037
BYTE32= 000040
BYTE33= 000041
BYTE34= 000042
BYTE35= 000043
BYTE36= 000044
BYTE37= 000045
BYTE38= 000046
BYTE39= 000047
BYTE4 = 000004
BYTE40= 000050
BYTE41= 000051
BYTE42= 000052
BYTE43= 000053
BYTE44= 000054
BYTE45= 000055
BYTE46= 000056
BYTE47= 000057
BYTE48= 000060
BYTE49= 000061
BYTE5 = 000005
BYTE50= 000062
BYTE51= 000063
BYTE52= 000064
BYTE53= 000065
BYTE54= 000066
BYTE55= 000067
BYTE56= 000070
BYTE57= 000071
BYTE58= 000072
BYTE59= 000073
BYTE6 = 000006
BYTE60= 000074
BYTE61= 000075
BYTE62= 000076
BYTE63= 000077
BYTE64= 000100
BYTE65= 000101
BYTE66= 000102
BYTE67= 000103
BYTE68= 000104
BYTE69= 000105
BYTE7 = 000007
BYTE70= 000106
BYTE71= 000107
BYTE72= 000110
BYTE73= 000111
BYTE74= 000112
BYTE75= 000113
BYTE76= 000114
BYTE77= 000115
BYTE78= 000116
BYTE79= 000117
BYTE8 = 000010
BYTE80= 000120
BYTE81= 000121
BYTE82= 000122
BYTE83= 000123
BYTE84= 000124
BYTE85= 000125
BYTE86= 000126
BYTE87= 000127
BYTE88= 000130
BYTE89= 000131
BYTE9 = 000011
BYTE90= 000132
BYTE91= 000133
BYTE92= 000134
BYTE93= 000135
BYTE94= 000136
BYTE95= 000137
BYTE96= 000140
BYTE97= 000141
BYTE98= 000142
BYTE99= 000143
BYTVAL= 000144
CBKALL= 001000
CBKCLK= 000400
CL    = ****** GX
CL1     001162RG    002
CNOBRE= 100000
CONSOL= ****** GX
CPCCEN= 010000
CPREAD= 040000
CPWRTE= 020000
CSADRD= 000004
CSEQCI= 100000
CSOE  = 000040
CSR1  = ****** GX
CSWRTE= 000100
DBR.RD= 000001
DB$CPP= 001457
DB$SPT= 000026
DB$TPC= 000023
DISPGS= 100000
DMAAWR= 000005
DMARRD= 000003
DMARWR= 000004
ENBR  = 010000
ERR4  = ****** GX
ERR5  = ****** GX
ERR6  = ****** GX
FIND  = ****** GX
KILL  = ****** GX
LBMRP = ****** GX
LOCTBL  000046R     002
LOCWRD  000044R     002
LOC.EN= 000100
LOC.WA= 040000
LOC.WB= 100000
LOC1P   001172RG    002
LOC1S   001176RG    002
LOC1W   001202RG    002
LOOP  = ****** GX
LOOPR = ****** GX
MAREN1= 000001
MAREN2= 004000
MARLOD= 010000
MAROUT= 000002
MAR.LO= 002000
MAR.OU= 000040
MBKALL= 001000
MBKCLK= 000400
MMADRD= 000100
MMLEFT= 000002
MMOE  = 000004
MMWRTE= 000010
MNOBRE= 100000
MREN1 = 000001
MREN2 = 020000
MRPCR = ****** GX
MRPLB = ****** GX
MRPXX = ****** GX
MSYN  = 000040
N     = 000144
ONCE  = ****** GX
PACK  = ****** GX
PLB   = 000010
PLC   = 000020
PLD   = 000030
PLRWR = 000200
PLR.EN= 000200
PRINT = ****** GX
QR$CR1= 176420
QR$CR2= 176422
QR$LBR= 176424
Q$ATTN= 000100
Q$BCL = 000001
Q$CCCP= 000040
Q$CHB = 000400
Q$CHRL= 000200
Q$CLR = 000040
Q$CNC = 030000
Q$CP  = 000060
Q$CPCC= 000010
Q$CP2 = 000260
Q$CSC = 010000
Q$CSEL= 000360
Q$CSET= 000002
Q$CSP = 020000
Q$DMA = 000001
Q$ENBK= 040000
Q$ENOP= 020000
Q$FAL = 004000
Q$FC  = 000045
Q$FO  = 000044
Q$FP  = 000046
Q$HBF = 000002
Q$ICP = 000006
Q$IHB = 000003
Q$IHRL= 000002
Q$IMRP= 000007
Q$LBD = 001000
Q$LBDP= 001001
Q$LBP = 000001
Q$LDCD= 000003
Q$LDMD= 000004
Q$LDPP= 002000
Q$LHP = 010000
Q$MNC = 140000
Q$MR  = 000052
Q$MRP = 000040
Q$MRP2= 000240
Q$MSC = 040000
Q$MSET= 000004
Q$MSP = 100000
Q$NCLK= 176000
Q$PP  = 000100
Q$PPSW= 000320
Q$PP2 = 000300
Q$QHLT= 000013
Q$QL  = 000043
Q$QLA = 000053
Q$QLB = 000054
Q$QLR = 000001
Q$QW  = 000042
Q$RDCD= 000005
Q$RDMD= 000006
Q$REBK= 001000
Q$RNC = 006000
Q$RSC = 004000
Q$RSET= 000010
Q$SM  = 100000
Q$SP  = 000120
Q$SP2 = 000340
REi     000500RG    002
REiDW   000736RG    002
REiIN   000554R     002
REiLN = 000004 G
REiMA   000564RG    002
REiMP   000612RG    002
REiPUT  001046R     002
REiSW   000644RG    002
REiTBL  000056RG    002
REiX    001114R     002
RGO.EN= 000200
RGO.VA= 020000
RSi     001132RG    002
RTNPT = ****** GX
SCAN  = ****** GX
SEQ.CI= 000010
STi     000076RG    002
STiBK   000230RG    002
STiCR   000216RG    002
STiIN   000206R     002
STiLN = 000011 G
STiLP   000454R     002
STiLPG  000446RG    002
STiLS   000440RG    002
STiLW   000432RG    002
STiMA   000252RG    002
STiPL   000314RG    002
STiPR   000336RG    002
STiQR   000360RG    002
STiTBL  000000RG    002
STiX    000470R     002
S$CLR = 000000
S$LA  = 000001
```

```
SYMBOL·TABLE·

S$QB·· =·000005          T$FSB2=·000010          WORD14=·000034          WORD44=·000130          WORD73=·000222
S$QR·  =·000006          T$IB·  =·000026          WORD15=·000036          WORD45=·000132·         WORD74=·000224
S$QX·  =·000004          T$IBAR=·000024          WORD16=·000040          WORD46=·000134          WORD75=·000226
S$SR·  =·000007          T$IBE·=·020000          WORD17=·000042·         WORD47=·000136          WORD76=·000230
S$S1   =·000010          T$IBF·=·040000          WORD18=·000044          WORD48=·000140          WORD77=·000232
S$S2·  =·000014          T$ICD·=·000040          WORD19=·000046          WORD49=·000142·         WORD78=·000234
TD$CTR=·176370           T$MODE=·004000          WORD2·=·000004          WORD5 =·000012·         WORD79=·000236
TD$CTW=·176360           T$OB·  =·000036          WORD20=·000050          WORD50=·000144          WORD8 =·000020
TD$INL=·004000           T$OBE·=·004000          WORD21=·000052·         WORD51=·000146          WORD80=·000240
TD$MEM=·000270           T$OBF·=·010000          WORD22=·000054          WORD52=·000150          WORD81=·000242
TD$OAR=·176344           T$OBRA=·000034          WORD23=·000056          WORD53=·000152·         WORD82=·000244
TD$OTR=·176346           T$OBWA=·000032·         WORD24=·000060          WORD54=·000154          WORD83=·000246
TD$QRD=·000274           T$OUTA=·100000          WORD25=·000062·         WORD55=·000156          WORD84=·000250
TD$SW·=·176376           T$RBDO=·000200          WORD26=·000064          WORD56=·000160          WORD85=·000252
TD$TAR=·176372·          T$RNB·=·000040          WORD27=·000066          WORD57=·000162·         WORD86=·000254
TD$TAW=·176362·          T$RSET=·040000          WORD28=·000070          WORD58=·000164          WORD87=·000256
TD$TDR=·176374           T$SC·  =·000022·         WORD29=·000072·         WORD59=·000166          WORD88=·000260
TD$TDW=·176364           T$SCLK=·020000          WORD3 =·000006          WORD6 =·000014          WORD89=·000262
T$AD·  =·000020          T$SEG1=·000000          WORD30=·000074          WORD60=·000170          WORD9 =·000022
T$BA·  =·000002·         T$SEG2=·000001          WORD31=·000076          WORD61=·000172·         WORD90=·000264
T$BD·  =·000010          T$SEG3=·000002·         WORD32=·000100          WORD62=·000174          WORD91=·000266
T$BSO·=·100000           T$SO·  =·000001          WORD33=·000102·         WORD63=·000176          WORD92=·000270
T$BT·  =·000020          T$UBUS=·100000          WORD34=·000104          WORD64=·000200          WORD93=·000272
T$BTAR=·000030           T$1CLK=·000400          WORD35=·000106          WORD65=·000202·         WORD94=·000274
T$BTD·=·002000           T$8BEN=·000020          WORD36=·000110          WORD66=·000204          WORD95=·000276
T$CD·  =·000100          UBD.IN=·000020          WORD37=·000112·         WORD67=·000206          WORD96=·000300
T$CLK·=·002000           UNPK·  =·******  GX·     WORD38=·000114          WORD68=·000210          WORD97=·000302
T$DISK=·000200           WORD0 =·000000          WORD39=·000116          WORD69=·000212·         WORD98=·000304
T$DRD·=·000004           WORD1 =·000002·         WORD4 =·000010          WORD7 =·000016          WORD99=·000306
T$EMEM=·010000           WORD10=·000024          WORD40=·000120          WORD70=·000214          WRDVAL=·000310
T$FSAA=·000000           WORD11=·000026          WORD41=·000122·         WORD71=·000216          XTREAD=·001000
T$FSAB=·000004           WORD12=·000030          WORD42=·000124          WORD72=·000220          XTWRTE=·000400
T$FSAC=·000014           WORD13=·000032·         WORD43=·000126

. ABS.  000000     000
        000000     001
MRREST· 001274     002·
ERRORS·DETECTED:  0

VIRTUAL·MEMORY·USED:  3285 WORDS·  (·13 PAGES)
DYNAMIC·MEMORY:  3860 WORDS·  ( 14 PAGES)
ELAPSED·TIME:  00:00:47
MRREST,MRREST/-SP=[20,1]IM,[20,1]MRREST·
```

```
                                        .TITLE  MRBUG
000000                                  .PSECT  MRBUG
                                ;
                                ;
                                ;       HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
                                ;       MATCH REPORT PROCESSOR TEST ROUTINES
                                ;
                                ;       DEBUGGING COMMANDS
                                ;       COMMANDS:
                                ;       AT      SET BREAKPOINT
                                ;       OF      REMOVE BREAKPOINT
                                ;       SS      SINGLE STEP
                                ;       GO      PROCEED FROM BREAKPOINT
                                ;
                                ;       ONCE A COMMAND HAS BEEN EXECUTED (OR AN ERROR ENCOUNTERED)
                                ;       THIS MODULE RETURNS CONTROL TO THE MODULE MRP AT LOCATION
                                ;       'MRPXX'.
                                ;
                                ;
                                ; **********************
                                ;
                                ;       DISCLAIMER:
                                ;       IN ORDER FOR THE BREAKPOINT ROUTINES TO WORK PROPERLY
                                ;       THERE MUST BE A 'PIPELINE REGISTER ONLY' CLOCK, THIS
                                ;       WOULD ALLOW THE CHANGING OF PLR CONTENTS WITHOUT CAUSING
                                ;       AN INSTRUCTION TO BE EXECUTED, AS IT STANDS NOW, THE PIPE-
                                ;       LINE REGISTER CLOCK IS TIED TO THE SEQUENCER CLOCK, WHEN
                                ;       A BREAKPOINT HAS BEEN REACHED, ANY OPERATION INVOLVING
                                ;       A PLR CLOCK (EG, PRINTING MICROPGM MEMORY) WILL EXECUTE
                                ;       THE INSTRUCTION IN THE PLR, THE 2900 SEQUENCER STACK IS
                                ;       ALSO SOMETIMES CHANGED.
                                ;
                                ; **********************
                                ;
                                ;
                                        .MCALL  CLEF$S,WTLO$S,RDAF$S,WTSE$S
                                ;
                                ;
                                ;       SET BREAKPOINT
                                ;       EG. MR>AT 22
                                ;
000000                          AT1::
000000                                  CALL    FIND                    ;LOOK FOR BKPT ADDRESS IN COMMAND LINE
000004  103004                         BCC     1$                      ;OK, CONVERT BKPT ADDRESS
000006                                  CALL    ERR4                    ;NOTHING THERE
000012  000167  000000G                 JMP     MRPXX
000016                          1$:     CALL    PACK                    ;CONVERT BKPT ADDRESS
000022  103004                         BCC     2$                      ;OK, CONTINUE
000024                                  CALL    ERR5                    ;ERROR ON CONVERSION
000030  000167  000000G                 JMP     MRPXX
000034  026767  000000G 000000G 2$:     CMP     MMHIGH,BINWD            ;IS ADDRESS IN RANGE
000042  003004                         BGT     3$                      ;YES, CONTINUE
000044                                  CALL    ERR10                   ;ADDR OUT OF RANGE
000050  000167  000000G                 JMP     MRPXX
                                ;
000054  012746  000400          3$:     MOV     #MBKCLKEN,-(SP)         ;SET MRP CNTL FOR BREAKPOINT
000060                                  CALL    MRPCR                   ;DIRECT CNTL WORD TO MRP
```

```
 58 000064  016746  000000G     MOV     BINWD,-(SP)             ;LOAD BKPT ADDR INTO LOAD BUS REG
 59 000070                      CALL    LBMRP                   ;LOAD MRP BKPT REG
 60 000074  052767  000000G 000000G BIS    #BREAK,BASE          ;SET FLAG FOR BKPT SET
 61 000102  000167  000000G     JMP     MRPXX
 62                             ;
 63                             ;
 64                             ;       REMOVE BREAKPOINT (DISABLE)
 65                             ;       MR>OF
 66                             ;
 67                             ;
 68 000106                      OF1::
 69 000106  005046              CLR     -(SP)                   ;SEND 0 TO MRP CONTROL REG
 70 000110                      CALL    MRPCR                   ;EXECUTE TRANSFER
 71 000114  005067  176422      CLR     QR$CR2                  ;CLEAR CSR 2
 72 000120  042767  000000G 000000G BIC    #BREAK,BASE          ;CLEAR BREAKPOINT SET FLAG
 73 000126  000167  000000G     JMP     MRPXX
 74                             ;
 75                             ;
 76                             ;       SINGLE STEP
 77                             ;       EG. MR>SS
 78                             ;           MR>SS 22
 79                             ;
 80                             ;
 81 000132                      SS1::
 82 000132                      CALL    FIND                    ;LOOK FOR ADDR IN COMMAND LINE
 83 000136  103426              BCS     3$                      ;NONE THERE, USE CURRENT MAR
 84 000140                      CALL    PACK                    ;CONVERT START ADDRESS
 85 000144  103004              BCC     1$                      ;OK, CONTINUE
 86 000146                      CALL    ERR5                    ;ERROR ON CONVERSION
 87 000152  000167  000000G     JMP     MRPXX
 88                             ;
 89 000156  026767  000000G 000000G 1$: CMP     MMHIGH,BINWD     ;IS ADDRESS IN RANGE
 90 000164  003004              BGT     2$                      ;YES, CONTINUE
 91 000166                      CALL    ERR10                   ;OUT OF RANGE HIGH
 92 000172  000167  000000G     JMP     MRPXX
 93                             ;
 94                             ;       SEQUENCE UP TO SINGLE STEP ADDRESS (IF NECESSARY)
 95                             ;       SET SEARCH MODE
 96                             ;       SINGLE CLOCK ALL TO EXECUTE ONE INSTRUCTION
 97                             ;       SET LOAD MODE
 98                             ;
 99 000176  016746  000000G     2$:     MOV     BINWD,-(SP)     ;PROVIDE START ADDRESS FOR SEQUENCER
100 000202                      CALL    SEQMM                   ;SEQUENCE ONLY UP TO START ADDRESS
101 000206  005046              CLR     -(SP)                   ;REINHIBIT BRANCH CONTROL REG
102 000210                      CALL    MRPCR
103 000214                      3$:
104 000214  012746  001777      MOV     #1777,-(SP)             ;SET CP MICROCODE TO INNOCUOUS LOC (X'3FF')
105 000220                      CALL    SEQCS
106 000224  005046              CLR     -(SP)                   ;REINHIBIT BR CONTROL REG
107 000226                      CALL    CPCR
108                            ;
109 000232  012767  100000 176422 MOV    #Q$SM,QR$CR2            ;SET SEARCH MODE
110 000240  012746  146000      MOV     #<Q$MNC+Q$RNC>,-(SP)    ;CLEAR MRP AND PPS 400-CLOCKS
111 000244  052716  001001      BIS     #<Q$LBD+Q$LBP>,(SP)     ;CLR DRIVE & PULSE
112 000250  012746  100000      MOV     #Q$MSP,-(SP)            ;SET SINGLE CLOCK ALL
113 000254                      CALL    CSR1                    ;MOVE TO CSR1
114 000260  005046              CLR     -(SP)                   ;CLEAR NOTHING
```

```
115 000262  012746  146000                  MOV     #<Q$MNC+Q$RNC>,-(SP)    ;SET MRP AND PPS NO-CLOCKS
116 000266                                  CALL    CSR1                    ;AND REWRITE CSR1
117                                 ;
118 000272  005067  176422                  CLR     QR$CR2                  ;SET LOAD MODE
119 000276                                  CALL    MRPMP                   ;PRINT MICROPGM ADDRESS
120 000302  000167  000000G                 JMP     MRPXX
121                                 ;
122                                 ;
123                                 ;       PROCEED
124                                 ;       EG. MR>GO
125                                 ;           MR>GO 22
126                                 ;
127                                 ;
128 000306                          GO1::
129 000306                                  CALL    FIND                    ;LOOK FOR START ADDRESS
130 000312  103431                          BCS     MGO                     ;NONE, GO FROM PRESENT ADDRESS
131 000314                                  CALL    PACK                    ;CONVERT START ADDRESS
132 000320  103004                          BCC     1$                      ;OK, CONTINUE
133 000322                                  CALL    ERR5                    ;ERROR ON CONVERSION
134 000326  000167  000000G                 JMP     MRPXX
135                                 ;
136 000332  026767  000000G 000000G 1$:     CMP     MMHIGH,BINWD            ;IS START ADDRESS IN RANGE
137 000340  003004                          BGT     2$                      ;YES, CONTINUE
138 000342                                  CALL    ERR10                   ;OUT OF RANGE HIGH
139 000346  000167  000000G                 JMP     MRPXX
140                                 ;
141 000352  016746  000000G         2$:     MOV     BINWD,-(SP)             ;PROVIDE START ADDRESS FOR SEQUENCER
142 000356                                  CALL    SEQMM                   ;SEQUENCE ONLY UP TO START ADDRESS
143                                 ;
144 000362  005046                          CLR     -(SP)                   ;REINHIBIT BRANCH CONTROL REG
145 000364                                  CALL    MRPCR
146 000370  005046                          CLR     -(SP)
147 000372                                  CALL    LBMRP                   ;SINGLE CLOCK TO REINHIBIT BRANCH REGISTER
148                                 ;
149                                 ;       IF A BREAKPOINT IS IN EFFECT, BRANCH TO CODE THAT PREPARES
150                                 ;       TO FIELD IT, ELSE START FREE RUN WITHOUT BREAKPOINTS.
151                                 ;
152 000376                          MGO:
153 000376  012746  001777                  MOV     #1777,-(SP)             ;SET CP MICROCODE TO INNOCUOUS LOC (X'3FF')
154 000402                                  CALL    SEQCS
155 000406  005046                          CLR     -(SP)                   ;REINHIBIT BR CONTROL REG
156 000410                                  CALL    CPCR
157                                 ;
158 000414  032767  000000G 000000G         BIT     #BREAK,BASE             ;PROCEED TO BREAKPOINT
159 000422  001047                          BNE     MBK                     ;YES, SET UP FOR BKPT
160 000424  012767  100000  176422          MOV     #Q$SM,QR$CR2            ;SET SEARCH MODE
161 000432  012746  146000                  MOV     #<Q$MNC+Q$RNC>,-(SP)    ;CLEAR MRP AND PPS NO-CLOCKS
162 000436  052716  001001                  BIS     #<Q$LBD+Q$LBP>,(SP)     ;CLEAR DRIVE AND PULSE
163 000442  005046                          CLR     -(SP)                   ;SET NOTHING
164 000444                                  CALL    CSR1                    ;NEW CSR1
165                                 ;
166                                 ;       PROVIDE FOR INTERRUPT FROM TERMINAL TO RETURN CONTROL
167                                 ;       TO USER (IN CASE MICROCODE HANGS OR RUNS FOREVER).
168                                 ;
169 000450                                  CLEF$S  #EFN.4                  ;CLEAR QIO EVENT FLAG
170                                 ;
171 000462                                  CALL    HANG2                   ;QIO FOR UNSOLICITED INTERRUPT
```

```
                                        WTSE$S  #EFN.4                  ;WAIT FOR QIO COMPLETION (CHAR INTERRUPT)
                                ;
                                        CALL    KILL                    ;KILL QIO
                                        CLR     -(SP)                   ;CLEAR NOTHING
                                        MOV     #<Q$MNC+Q$RNC>,-(SP)    ;SET MRP AND PPS NO-CLOCKS
                                        CALL    CSR1                    ;REWRITE CSR1
                                        CLR     -(SP)                   ;SEND 0 TO MRP CONTROL REG
                                        CALL    MRPCR                   ;DO IT
                                        MOV     #0,@R$CR2               ;SET LOAD MODE
                                        CALL    MRPMP                   ;PRINT MICROPGM ADDRESS
                                        JMP     MRPXX
                                ;
                                ;
                                ;       EXPECT TO HIT A BREAKPOINT.
                                ;
                                MBK:
                                        CLEF$S  #EFN.3                  ;CLEAR BKPT EVENT FLAG
                                ;
                                        CLEF$S  #EFN.4                  ;CLEAR QIO EVENT FLAG
                                ;
                                ;       SET BKPT ALLOW BIT IN MRP CONTROL REG
                                ;       SET SEARCH MODE
                                ;       SET FREE RUN CLOCK
                                ;       WAIT FOR BKPT INTERRUPT THROUGH VECTOR ADDRESS 274 (SEE
                                ;       RTN 'BPTISR') OR CHARACTER INTERRUPT FROM TERMINAL
                                ;
                                        MOV     #MBKALLOW,-(SP)         ;SET MRP CNTL WORD
                                        CALL    MRPCR                   ;DIRECT CNTL WORD TO MRP
                                        MOV     #<Q$SM+Q$ENBK>,@R$CR2   ;SET SEARCH MODE
                                        MOV     #<Q$MNC+Q$RNC>,-(SP)    ;CLEAR MRP AND PPS NO-CLOCKS
                                        BIS     #<Q$LBD+Q$LBP>,(SP)     ;CLEAR DRIVE AND PULSE
                                        CLR     -(SP)                   ;SET NOTHING
                                        CALL    CSR1                    ;NEW CSR1
                                        CALL    HANG2                   ;ISSUE QIO TO TERMINAL
                                ;
                                ;       WAIT FOR INTERRUPT FROM TERMINAL OR BREAKPOINT INTERRUPT
                                ;       FROM MRP.
                                ;
                                        WTLO$S  0,#000014               ;WAIT FOR BKPT OR CHAR INTERRUPT
                                ;
                                        RDAF$S  #EFBUF                  ;READ FOR DEBUGGING
                                ;
                                ;       INTERRUPT RECEIVED
                                ;       RESET NO-CLOCKS IN CSR #1
                                ;       SET LOAD MODE
                                ;       PRINT MICROPGM ADDRESS
                                ;
                                        CLR     -(SP)                   ;CLEAR NOTHING
                                        MOV     #<Q$MNC+Q$RNC>,-(SP)    ;SET MRP AND PPS NO-CLOCKS
                                        CALL    CSR1                    ;REWRITE CSR1
                                        MOV     #0,-(SP)                ;SEND 0 TO MRP CONTROL REG
                                        CALL    MRPCR                   ;DO IT
                                ;
                                        MOV     #0,@R$CR2               ;SET LOAD MODE
                                        MOV     #@$REBK,@R$CR2          ;RE-ARM INTERRUPTS
                                        CALL    MRPMP                   ;PRINT MICROPGM ADDRESS
                                        CALL    KILL                    ;KILL QIO
```

```
    229 000722· 000167  000000G·              JMP·     MRPXX·
```

```
                                ;
                                ;
                                ;       PRINT MICROPROGRAM ADDRESS
                                ;
000726                          MRPMP:
000726  012746  000100                  MOV     #MMADRD,-(SP)           ;READ MM ADDRESS SELECT BIT
000732                          CALL    MRPCR                           ;WRITE MRP CONTROL REGISTER
000736                          CALL    MRPLB                           ;NOW GET WORD FROM MRP
000742  012601                  MOV     (SP)+,R1                        ;LOAD R1 FOR CONVERSION
000744  042701  177400          BIC     #177400,R1                      ;CLEAR UPPER 6 BITS
000750  012705  000000G         MOV     #PRINT,R5                       ;R5 -> PRINT LINE
000754                          CALL    UNPK                            ;CONVERT MPA
000760                          CALL    CONSOL                          ;PRINT ON CONSOLE
000764                          RETURN
                                ;
000001                          .END
```

```
ALUCKE= 040000          BYTE39= 000047          BYTE90= 000132          MMADRD= 000100          Q$LHP = 010000
ALUOE = 004000          BYTE4 = 000004          BYTE91= 000133          MMHIGH= ****** GX       Q$MNC = 140000
AT1     000000RG  002   BYTE40= 000050          BYTE92= 000134          MMLEFT= 000002          Q$MR  = 000052
A01   = 010000          BYTE41= 000051          BYTE93= 000135          MMOE  = 000004          Q$MRP = 000040
BASE  = ****** GX       BYTE42= 000052          BYTE94= 000136          MMWRTE= 000010          Q$MRP2= 000240
BINWD = ****** GX       BYTE43= 000053          BYTE95= 000137          MNOBRE= 100000          Q$MSC = 040000
BITVAL= 000000          BYTE44= 000054          BYTE96= 000140          MREN1 = 000001          Q$MSET= 000004
BIT0  = 000001          BYTE45= 000055          BYTE97= 000141          MREN2 = 020000          Q$MSP = 100000
BIT1  = 000002          BYTE46= 000056          BYTE98= 000142          MRPCR = ****** GX       Q$NCLK= 176000
BIT10 = 002000          BYTE47= 000057          BYTE99= 000143          MRPLB = ****** GX       Q$PP  = 000100
BIT11 = 004000          BYTE48= 000060          BYTVAL= 000144          MRPMP   000726R   002   Q$PPSW= 000320
BIT12 = 010000          BYTE49= 000061          CBKALL= 001000          MRPXX = ****** GX       Q$PP2 = 000300
BIT13 = 020000          BYTE5 = 000005          CBKCLK= 000400          MSYN  = 000040          Q$QHLT= 000013
BIT14 = 040000          BYTE50= 000062          CNOBRE= 100000          N     = 000144          Q$QL  = 000043
BIT15 = 100000          BYTE51= 000063          CONSOL= ****** GX       OF1     000106RG  002   Q$QLA = 000053
BIT2  = 000004          BYTE52= 000064          CPCCEN= 010000          PACK  = ****** GX       Q$QLB = 000054
BIT3  = 000010          BYTE53= 000065          CPCR  = ****** GX       PLB   = 000010          Q$QLR = 000001
BIT4  = 000020          BYTE54= 000066          CPREAD= 040000          PLC   = 000020          Q$QW  = 000042
BIT5  = 000040          BYTE55= 000067          CPWRTE= 020000          PLD   = 000030          Q$RDCD= 000005
BIT6  = 000100          BYTE56= 000070          CSADRD= 000004          PLRWR = 000200          Q$RDMD= 000006
BIT7  = 000200          BYTE57= 000071          CSEQCI= 100000          PLR.EN= 000200          Q$REBK= 001000
BIT8  = 000400          BYTE58= 000072          CSOE  = 000040          PRINT = ****** GX       Q$RNC = 006000
BIT9  = 001000          BYTE59= 000073          CSR1  = ****** GX       QR$CR1= 176420          Q$RSC = 004000
BREAK = ****** GX       BYTE6 = 000006          CSWRTE= 000100          QR$CR2= 176422          Q$RSET= 000010
BYTE0 = 000000          BYTE60= 000074          DBR.RD= 000001          QR$LBR= 176424          Q$SM  = 100000
BYTE1 = 000001          BYTE61= 000075          DB$CPP= 001457          Q$ATTN= 000100          Q$SP  = 000120
BYTE10= 000012          BYTE62= 000076          DB$SPT= 000026          Q$BCL = 000001          Q$SP2 = 000340
BYTE11= 000013          BYTE63= 000077          DB$TPC= 000023          Q$CCCP= 000040          RGQ.EN= 000200
BYTE12= 000014          BYTE64= 000100          DISPGS= 100000          Q$CHB = 000400          RGQ.VA= 020000
BYTE13= 000015          BYTE65= 000101          DMAAWR= 000005          Q$CHRL= 000200          SEQCS = ****** GX
BYTE14= 000016          BYTE66= 000102          DMARRD= 000003          Q$CLR = 000040          SEQMM = ****** GX
BYTE15= 000017          BYTE67= 000103          DMARWR= 000004          Q$CNC = 030000          SEQ.CI= 000010
BYTE16= 000020          BYTE68= 000104          EFBUF = ****** GX       Q$CP  = 000060          SSI     000132RG  002
BYTE17= 000021          BYTE69= 000105          EFN.3 = ****** GX       Q$CPCC= 000010          S$CLR = 000000
BYTE18= 000022          BYTE7 = 000007          EFN.4 = ****** GX       Q$CP2 = 000260          S$LA  = 000001
BYTE19= 000023          BYTE70= 000106          ENBR  = 010000          Q$CSC = 010000          S$QB  = 000005
BYTE2 = 000002          BYTE71= 000107          ERR10 = ****** GX       Q$CSEL= 000360          S$QR  = 000006
BYTE20= 000024          BYTE72= 000110          ERR4  = ****** GX       Q$CSET= 000002          S$QX  = 000004
BYTE21= 000025          BYTE73= 000111          ERR5  = ****** GX       Q$CSP = 020000          S$SR  = 000007
BYTE22= 000026          BYTE74= 000112          FIND  = ****** GX       Q$DMA = 000001          S$S1  = 000010
BYTE23= 000027          BYTE75= 000113          GO1     000306RG        Q$ENBK= 040000  002     S$S2  = 000014
BYTE24= 000030          BYTE76= 000114          HANG2 = ****** GX       Q$ENOP= 020000          TD$CTR= 176370
BYTE25= 000031          BYTE77= 000115          KILL  = ****** GX       Q$FAL = 004000          TD$CTW= 176360
BYTE26= 000032          BYTE78= 000116          LBMRP = ****** GX       Q$FC  = 000045          TD$INL= 004000
BYTE27= 000033          BYTE79= 000117          LOC.EN= 000100          Q$FO  = 000044          TD$MEM= 000270
BYTE28= 000034          BYTE8 = 000010          LOC.WA= 040000          Q$FP  = 000046          TD$OAR= 176344
BYTE29= 000035          BYTE80= 000120          LOC.WB= 100000          Q$HBF = 000002          TD$OTR= 176346
BYTE3 = 000003          BYTE81= 000121          MAREN1= 000001          Q$ICP = 000006          TD$ORD= 000274
BYTE30= 000036          BYTE82= 000122          MAREN2= 004000          Q$IHB = 000003          TD$SW = 176376
BYTE31= 000037          BYTE83= 000123          MARLOD= 010000          Q$IHRL= 000002          TD$TAR= 176372
BYTE32= 000040          BYTE84= 000124          MAROUT= 000002          Q$IMRP= 000007          TD$TAW= 176362
BYTE33= 000041          BYTE85= 000125          MAR.LO= 002000          Q$LBD = 001000          TD$TDR= 176374
BYTE34= 000042          BYTE86= 000126          MAR.OU= 000040          Q$LBDP= 001001          TD$TDW= 176364
BYTE35= 000043          BYTE87= 000127          MBK     000542R   002   Q$LBP = 000001          T$BA  = 000002
BYTE36= 000044          BYTE88= 000130          MBKALL= 001000          Q$LDCD= 000003          T$BD  = 000010
BYTE37= 000045          BYTE89= 000131          MBKCLK= 000400          Q$LDMD= 000004          T$BSO = 100000
BYTE38= 000046          BYTE9 = 000011          MGO     000376R   002   Q$LDPP= 002000
```

```
MRBUG  MACRO M1110  27-MAR-80 15:09  PAGE 6-2
SYMBOL TABLE
```

```
T$BT  = 000020          T$SEG1 = 000000          WORD26 = 000064          WORD51 = 000146          WORD77 = 000232
T$BTAR = 000030         T$SEG2 = 000001          WORD27 = 000066          WORD52 = 000150          WORD78 = 000234
T$BTD  = 002000         T$SEG3 = 000002          WORD28 = 000070          WORD53 = 000152          WORD79 = 000236
T$CD  = 000100          T$SO  = 000001           WORD29 = 000072          WORD54 = 000154          WORD8 = 000020
T$CLK  = 002000         T$UBUS = 100000          WORD3 = 000006           WORD55 = 000156          WORD80 = 000240
T$DISK = 000200         T$1CLK = 000400          WORD30 = 000074          WORD56 = 000160          WORD81 = 000242
T$DRD  = 000004         T$8BEN = 000020          WORD31 = 000076          WORD57 = 000162          WORD82 = 000244
T$EMEM = 010000         UBD.IN = 000020          WORD32 = 000100          WORD58 = 000164          WORD83 = 000246
T$FSAA = 000000         UNPK  = ****** GX        WORD33 = 000102          WORD59 = 000166          WORD84 = 000250
T$FSAB = 000004         WORD0 = 000000           WORD34 = 000104          WORD6 = 000014           WORD85 = 000252
T$FSAC = 000014         WORD1 = 000002           WORD35 = 000106          WORD60 = 000170          WORD86 = 000254
T$FSB2 = 000010         WORD10 = 000024          WORD36 = 000110          WORD61 = 000172          WORD87 = 000256
T$IB  = 000026          WORD11 = 000026          WORD37 = 000112          WORD62 = 000174          WORD88 = 000260
T$IBAR = 000024         WORD12 = 000030          WORD38 = 000114          WORD63 = 000176          WORD89 = 000262
T$IBE  = 020000         WORD13 = 000032          WORD39 = 000116          WORD64 = 000200          WORD9 = 000022
T$IBF  = 040000         WORD14 = 000034          WORD4 = 000010           WORD65 = 000202          WORD90 = 000264
T$ICD  = 000040         WORD15 = 000036          WORD40 = 000120          WORD66 = 000204          WORD91 = 000266
T$MODE = 004000         WORD16 = 000040          WORD41 = 000122          WORD67 = 000206          WORD92 = 000270
T$OB  = 000036          WORD17 = 000042          WORD42 = 000124          WORD68 = 000210          WORD93 = 000272
T$OBE  = 004000         WORD18 = 000044          WORD43 = 000126          WORD69 = 000212          WORD94 = 000274
T$OBF  = 010000         WORD19 = 000046          WORD44 = 000130          WORD7 = 000016           WORD95 = 000276
T$OBRA = 000034         WORD2 = 000004           WORD45 = 000132          WORD70 = 000214          WORD96 = 000300
T$OBWA = 000032         WORD20 = 000050          WORD46 = 000134          WORD71 = 000216          WORD97 = 000302
T$OUTA = 100000         WORD21 = 000052          WORD47 = 000136          WORD72 = 000220          WORD98 = 000304
T$RBDO = 000200         WORD22 = 000054          WORD48 = 000140          WORD73 = 000222          WORD99 = 000306
T$RNB  = 000040         WORD23 = 000056          WORD49 = 000142          WORD74 = 000224          WRDVAL = 000310
T$RSET = 040000         WORD24 = 000060          WORD5 = 000012           WORD75 = 000226          XTREAD = 001000
T$SC  = 000022          WORD25 = 000062          WORD50 = 000144          WORD76 = 000230          XTWRTE = 000400
T$SCLK = 020000
```

```
. ABS.  000000     000
        000000     001
MRBUG   000766     002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3584 WORDS  ( 14 PAGES)
DYNAMIC MEMORY:  4916 WORDS  ( 18 PAGES)
ELAPSED TIME:  00:00:48
MRBUG,MRBUG/-SP=[20,1]JIM,[20,1]MRBUG
```

```
000000                      .TITLE GR
                            .PSECT CP
                            .LIST MEB
                  ;
                  ;
                  ;         HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
                  ;         CONTROL PROCESSOR TEST ROUTINES
                  ;
                  ;         PARSE THE COMMAND LINE AND TRANSFER CONTROL TO ONE OF THE
                  ;         CP SUB-MODULES OR BACK TO QMAIN. CONTROL COULD HAVE BEEN
                  ;         PASSED TO CP IN ONE OF THREE WAYS:
                  ;         1. FROM QMAIN IF CP WAS SELECTED AS THE FIRST PROCESSOR
                  ;            UPON ENTRY TO THE PROGRAM
                  ;            >CP LD CD 0
                  ;         2. FROM QMAIN IF CP WAS SELECTED FROM ONE OF QMAIN'S
                  ;            OTHER SUB-MODULES
                  ;            MR>CP LD CD 0
                  ;         3. ON RETURN FROM ONE OF CP'S SUB-MODULES
                  ;            CP>LD CD 0
                  ;
                  ;         CP PARSES AT THE SECOND LEVEL OF CONTROL (SEE NOTES AT
                  ;         QMAIN ON LEVELS OF CONTROL). IN THE EXAMPLES ABOVE, CP
                  ;         WOULD PARSE 'LD' AND TRANSFER CONTROL TO THE CP SUB-
                  ;         MODULE CPLD. IF CP SHOULD ENCOUNTER A STRING WHICH IS
                  ;         NOT A VALID COMMAND MNEMONIC, CP CONSIDERS THE STRING TO
                  ;         BE A PROCESSOR MNEMONIC AND RETURNS CONTROL TO QMAIN. EG:
                  ;         CP>MR LD MM 0
                  ;         THE STRING 'MR' IS NOT A VALID CP (SECOND LEVEL) COMMAND.
                  ;         CP RETURNS CONTROL TO QMAIN WHICH IN TURN WILL TRANSFER
                  ;         CONTROL TO ITS SUB-MODULE MRP.
                  ;
                  ;         CP SUB-MODULES:
                  ;         CP_D      LOAD MEMORIES
                  ;         CPPR      PRINT MEMORY CONTENTS
                  ;         CPREST    REMAINING COMMANDS (EXCEPT DEBUGGING COMMANDS)
                  ;         CPBUG1    DEBUGGING COMMANDS
                  ;         CPBUG2    DEBUGGING COMMANDS
```

```
                                        ;
                                        ;
                                        ;       TABLE OF CP COMMAND MNEMONICS AND THEIR ASSOCIATED ROUTINE
                                        ;       ADDRESSES
                                        ;
                                        ;
000000                                  CPTBL:
000000      101     124                         .ASCII  /AT/            ;SET BREAKPOINT
000002  000000G                                 .WORD   AT2
000004      117     106                         .ASCII  /OF/            ;DISABLE BREAKPOINT
000006  000000G                                 .WORD   OF2
000010      114     122                         .ASCII  /LR/            ;LOAD ALL REGISTERS
000012  000000G                                 .WORD   LR2
000014      122     107                         .ASCII  /RG/            ;PRINT REGISTERS
000016  000000G                                 .WORD   RG2
000020      120     103                         .ASCII  /PC/            ;SINGLE STEP A NUMBER OF TIMES (PROCEED)
000022  000000G                                 .WORD   PC2
000024      123     123                         .ASCII  /SS/            ;SINGLE STEP
000026  000000G                                 .WORD   SS2
000030      107     117                         .ASCII  /GO/            ;PROCEED FROM BREAKPOINT
000032  000000G                                 .WORD   GO2
000034      106     122                         .ASCII  /FR/            ;RUN BOTH MRP AND CP
000036  000000G                                 .WORD   FR2
000040      123     124                         .ASCII  /ST/            ;STORE INTO A REG
000042  000000G                                 .WORD   ST2
000044      122     105                         .ASCII  /RE/            ;READ FROM A REG
000046  000000G                                 .WORD   RE2
000050      114     104                         .ASCII  /LD/            ;LOAD MEMORY
000052  000000G                                 .WORD   LD2
000054      120     122                         .ASCII  /PR/            ;PRINT FROM MEMORY
000056  000000G                                 .WORD   PR2
000060      120     102                         .ASCII  /PB/            ;PRINT BUFFERS
000062  000000G                                 .WORD   PB2
000064      122     123                         .ASCII  /RS/
000066  000000G                                 .WORD   RS2
000070      103     114                         .ASCII  /CL/            ;CALL HQR LOADER
000072  000000G                                 .WORD   CL2
          000017                        CPNUM   =       <.-CPTBL>/4
```

```
                                        ;
                                        ;       INDEX WORD FOR READING AND WRITING CP REGISTERS
                                        ;
000074  000000                 INDEX:: .WORD   0
                                        ;
                                        ;       BCL
                                        ;
000076                          BCL::
000076  000200                          .WORD   128.            ;BUFFERS ARE 128 WORDS
000100  000130'                         .WORD   HLB             ;HIT LIST BUFFER
000102  000000                          .WORD   0
000104  000000                          .WORD   0
000106  000000                          .WORD   0
000110  000530'                         .WORD   HRL             ;HRL BUFFER
000112  000000                          .WORD   0
000114  000000                          .WORD   0
000116  000000                          .WORD   0
000120                                  .BLKW   4               ;END OF BUFFER ADDRESSES
                                        ;
                                        ;       HIT LIST AND HRL BUFFERS
                                        ;
000130                          HLB::
000130                                  .BLKW   128.
000530                          HRL::
000530  000001                          .WORD   000001          ;=X'0001'
000532  021042                          .WORD   021042          ;=X'2222'
000534  031464                          .WORD   031464          ;=X'3334'
000536  000401                          .WORD   000401          ;=X'0101'
                                        ;
000540  000004                          .WORD   000004          ;=X'0004'
000542  052525                          .WORD   052525          ;=X'5555'
000544  063146                          .WORD   063146          ;=X'6666'
000546  000402                          .WORD   000402          ;=X'0102'
000550  000002                          .WORD   000002          ;=X'0002'
                                        ;
000552  000007                          .WORD   000007          ;=X'0007'
000554  104210                          .WORD   104210          ;=X'8888'
000556  114631                          .WORD   114631          ;=X'9999'
000560  000403                          .WORD   000403          ;=X'0103'
000562  001402                          .WORD   001402          ;=X'0302'
                                        ;
000564  000012                          .WORD   000012          ;=X'000A'
000566  135673                          .WORD   135673          ;=X'BBBB'
000570  146314                          .WORD   146314          ;=X'CCCC'
000572  000401                          .WORD   000401          ;=X'0101'
000574  140000                          .WORD   140000          ;=X'C000'  ;EOC
000576                                  .BLKW   128.-<.-HRL/2>
                                        .LIST   BEX
                                        .NLIST  CND
```

```
                                        ;
                                        ;
                                        ;       CP ROUTINES
                                        ;
                                        ;
001130                                  CP::
001130  004767  000000G                         JSR     PC,FIND         ;LOCATE THE COMMAND IN THE COMMAND LINE
001134  103003                                  BCC     1$              ;OK, CONTINUE
001136  004767  000000G                         JSR     PC,ERR3
001142  000422                                  BR      CPXX
001144  022700  000002          1$:             CMP     #2,R0           ;COMMANDS ARE 2 CHARS
001150  001403                                  BEQ     2$
001152  004767  000000G                         JSR     PC,ERR8
001156  000414                                  BR      CPXX            ;TRY AGAIN
                                        ;
001160  012700  000017          2$:             MOV     #CPNUM,R0       ;R0 = NUMBER OF COMMANDS
001164  012702  000000'                         MOV     #CPTBL,R2       ;R2 -> TABLE OF COMMAND MNEMONICS
001170  004767  000000G                         JSR     PC,SCAN         ;FIND MATCH IN TABLE
001174  103003                                  BCC     3$              ;OK, CONTINUE
001176  004767  000000G                         JSR     PC,ERR12        ;COMMAND NOT IN TABLE
001202  000402                                  BR      CPXX            ;TRY AGAIN
                                        ;
                                        ;       JUMP TO THE ROUTINE THAT GOVERNS THE COMMAND
                                        ;
001204  000171  000000          3$:             JMP     @(R1)
                                        ;
                                        ;
                                        ;       LOCAL CP LOOP, INCLUDE PROMPT FOR CP COMMAND.
                                        ;
                                        ;
001210                                  CPXX::
001210  012767  050103  000000C                 MOV     #"CP,GCMBLK+G.DPRM+2    ;MOVE CP NAME TO GCM BLOCK
001216  004767  000000G                         JSR     PC,GCONLY       ;PROMPT
001222  004767  000000G                         JSR     PC,FIND         ;LOCATE THE COMMAND IN THE COMMAND LINE
001226  103003                                  BCC     1$              ;OK, CONTINUE
001230  004767  000000G                         JSR     PC,ERR3
001234  000765                                  BR      CPXX
001236  022700  000002          1$:             CMP     #2,R0           ;COMMANDS ARE 2 CHARS
001242  001403                                  BEQ     2$
001244  004767  000000G                         JSR     PC,ERR8
001250  000757                                  BR      CPXX            ;TRY AGAIN
                                        ;
001252  012700  000017          2$:             MOV     #CPNUM,R0       ;R0 = NUMBER OF COMMANDS
001256  012702  000000'                         MOV     #CPTBL,R2       ;R2 -> TABLE OF COMMAND MNEMONICS
001262  004767  000000G                         JSR     PC,SCAN         ;FIND MATCH IN TABLE
001266  103005                                  BCC     3$              ;OK, CONTINUE
001270  042767  000000G 000000G                 BIC     #BREAK,BASE     ;CLEAR BREAKPOINT FLAG
001276  000167  000000G                         JMP     COMXX           ;RETURN TO QMAIN, LOOK FOR PROCESSOR MNEMONIC,
                                        ;
                                        ;       JUMP TO THE ROUTINE THAT GOVERNS THE COMMAND
                                        ;
001302  000171  000000          3$:             JMP     @(R1)
```

```
                                ;
                                ;
                                ; **********************
                                ;
                                ;       TEMPORARILY DELETED
                                ;
                                ; **********************
                                ;
                                ;
                                ;       LOAD A CP INTERNAL REGISTER
                                ;
                                ;       UPON ENTRY:
                                ;       R0 = REGISTER NUMBER
                                ;       THE FIELD 'BINWD' CONTAINS THE BIT PATTERN TO BE LOADED
                                ;
                                ;
001306                          LDREG::
001306  000207                          RTS     PC
```

```
                                ;
                                ;
                                ; **********************
                                ;
                                ;       TEMPORARILY DELETED
                                ;
                                ; **********************
                                ;
                                ;
                                ;       PRINT CP REGISTERS
                                ;
                                ;       UPON ENTRY:
                                ;       R0 = REGISTER TO BE READ
                                ;
                                ;       UPON EXIT:
                                ;       R1 = REGISTER VALUE
                                ;
                                ;
001310                          REREG::
001310  000207                          RTS     PC
                                ;
000001                                  .END
```

```
ALUCKE= 040000          BYTE39= 000047          BYTE90= 000132          MAROUT= 000002          Q$LDPP= 002000
ALUOE = 004000          BYTE4 = 000004          BYTE91= 000133          MAR.LO= 002000          Q$LHP = 010000
AT2   = ****** GX       BYTE40= 000050          BYTE92= 000134          MAR.OU= 000040          Q$MNC = 140000
A01   = 010000          BYTE41= 000051          BYTE93= 000135          MBKALL= 001000          Q$MR  = 000052
BASE  = ****** GX       BYTE42= 000052          BYTE94= 000136          MBKCLK= 000400          Q$MRP = 000040
BCL     000076RG  002   BYTE43= 000053          BYTE95= 000137          MMADRD= 000100          Q$MRP2= 000240
BITVAL= 000000          BYTE44= 000054          BYTE96= 000140          MMLEFT= 000002          Q$MSC = 040000
BIT0  = 000001          BYTE45= 000055          BYTE97= 000141          MMOE  = 000004          Q$MSET= 000004
BIT1  = 000002          BYTE46= 000056          BYTE98= 000142          MMWRTE= 000010          Q$MSP = 100000
BIT10 = 002000          BYTE47= 000057          BYTE99= 000143          MNOBRE= 100000          Q$NCLK= 176000
BIT11 = 004000          BYTE48= 000060          BYTVAL= 000144          MREN1 = 000001          Q$PP  = 000100
BIT12 = 010000          BYTE49= 000061          CBKALL= 001000          MREN2 = 020000          Q$PPSW= 000320
BIT13 = 020000          BYTE5 = 000005          CBKCLK= 000400          MSYN  = 000040          Q$PP2 = 000300
BIT14 = 040000          BYTE50= 000062          CL2   = ****** GX       N     = 000144          Q$QHLT= 000013
BIT15 = 100000          BYTE51= 000063          CNOBRE= 100000          OF2   = ****** GX       Q$QL  = 000043
BIT2  = 000004          BYTE52= 000064          COMXX = ****** GX       PB2   = ****** GX       Q$QLA = 000053
BIT3  = 000010          BYTE53= 000065          CP      001130RG  002   PC2   = ****** GX       Q$QLB = 000054
BIT4  = 000020          BYTE54= 000066          CPCCEN= 010000          PLB   = 000010          Q$QLR = 000001
BIT5  = 000040          BYTE55= 000067          CPNUM = 000017          PLC   = 000020          Q$QW  = 000042
BIT6  = 000100          BYTE56= 000070          CPREAD= 040000          PLD   = 000030          Q$RDCD= 000005
BIT7  = 000200          BYTE57= 000071          CPTBL   000000R   002   PLRWR = 000200          Q$RDMD= 000006
BIT8  = 000400          BYTE58= 000072          CPWRTE= 020000          PLR.EN= 000200          Q$REBK= 001000
BIT9  = 001000          BYTE59= 000073          CPXX    001210RG  002   PR2   = ****** GX       Q$RNC = 006000
BREAK = ****** GX       BYTE6 = 000006          CSADRD= 000004          QR$CR1= 176420          Q$RSC = 004000
BYTE0 = 000000          BYTE60= 000074          CSEQCI= 100000          QR$CR2= 176422          Q$RSET= 000010
BYTE1 = 000001          BYTE61= 000075          CSQE  = 000040          QR$LBR= 176424          Q$SM  = 100000
BYTE10= 000012          BYTE62= 000076          CSWRTE= 000100          Q$ATTN= 000100          Q$SP  = 000120
BYTE11= 000013          BYTE63= 000077          DBR.RD= 000001          Q$BCL = 000001          Q$SP2 = 000340
BYTE12= 000014          BYTE64= 000100          DB$CPP= 001457          Q$CCCP= 000040          REREG   001310RG  002
BYTE13= 000015          BYTE65= 000101          DB$SPT= 000026          Q$CHB = 000400          RE2   = ****** GX
BYTE14= 000016          BYTE66= 000102          DB$TPC= 000023          Q$CHRL= 000200          RGQ.EN= 000200
BYTE15= 000017          BYTE67= 000103          DISPGS= 100000          Q$CLR = 000040          RGQ.VA= 020000
BYTE16= 000020          BYTE68= 000104          DMAAWR= 000005          Q$CNC = 030000          RG2   = ****** GX
BYTE17= 000021          BYTE69= 000105          DMARRD= 000003          Q$CP  = 000060          RS2   = ****** GX
BYTE18= 000022          BYTE7 = 000007          DMARWR= 000004          Q$CPCC= 000010          SCAN  = ****** GX
BYTE19= 000023          BYTE70= 000106          ENBR  = 010000          Q$CP2 = 000260          SEQ.CI= 000010
BYTE2 = 000002          BYTE71= 000107          ERR12 = ****** GX       Q$CSC = 010000          SS2   = ****** GX
BYTE20= 000024          BYTE72= 000110          ERR3  = ****** GX       Q$CSEL= 000360          ST2   = ****** GX
BYTE21= 000025          BYTE73= 000111          ERR8  = ****** GX       Q$CSET= 000002          S$CLR = 000000
BYTE22= 000026          BYTE74= 000112          FIND  = ****** GX       Q$CSP = 020000          S$LA  = 000001
BYTE23= 000027          BYTE75= 000113          FR2   = ****** GX       Q$DMA = 000001          S$QB  = 000005
BYTE24= 000030          BYTE76= 000114          GCMBLK= ****** GX       Q$ENBK= 040000          S$QR  = 000006
BYTE25= 000031          BYTE77= 000115          GCONLY= ****** GX       Q$ENOP= 020000          S$QX  = 000004
BYTE26= 000032          BYTE78= 000116          GO2   = ****** GX       Q$FAL = 004000          S$SR  = 000007
BYTE27= 000033          BYTE79= 000117          G.DPRM= ****** GX       Q$FC  = 000045          S$S1  = 000010
BYTE28= 000034          BYTE8 = 000010          HLB     000130RG  002   Q$FO  = 000044          S$S2  = 000014
BYTE29= 000035          BYTE80= 000120          HR      000530RG  002   Q$FP  = 000046          TD$CTR= 176370
BYTE3 = 000003          BYTE81= 000121          INDEX   000074RG  002   Q$HBF = 000002          TD$CTW= 176360
BYTE30= 000036          BYTE82= 000122          LDREG   001306RG  002   Q$ICP = 000006          TD$INL= 004000
BYTE31= 000037          BYTE83= 000123          LD2   = ****** GX       Q$IHB = 000003          TD$MEM= 000270
BYTE32= 000040          BYTE84= 000124          LOC.EN= 000100          Q$IHRL= 000002          TD$OAR= 176344
BYTE33= 000041          BYTE85= 000125          LOC.WA= 040000          Q$IMRP= 000007          TD$OTR= 176346
BYTE34= 000042          BYTE86= 000126          LOC.WB= 100000          Q$LBD = 001000          TD$QRD= 000274
BYTE35= 000043          BYTE87= 000127          LR2   = ****** GX       Q$LBDP= 001001          TD$SW = 176376
BYTE36= 000044          BYTE88= 000130          MAREN1= 000001          Q$LBP = 000001          TD$TAR= 176372
BYTE37= 000045          BYTE89= 000131          MAREN2= 004000          Q$LDCD= 000003          TD$TAW= 176362
BYTE38= 000046          BYTE9 = 000011          MARLOD= 010000          Q$LDMD= 000004          TD$TDR= 176374
```

SYMBOL TABLE

```
TD$TDW= 176364          T$RBDO= 000200          WORD23= 000056          WORD5 = 000012          WORD76= 000230
T$AD  = 000020          T$RNB = 000040          WORD24= 000060          WORD50= 000144          WORD77= 000232
T$BA  = 000002          T$RSET= 040000          WORD25= 000062          WORD51= 000146          WORD78= 000234
T$BD  = 000010          T$SC  = 000022          WORD26= 000064          WORD52= 000150          WORD79= 000236
T$BSO = 100000          T$SCLK= 020000          WORD27= 000066          WORD53= 000152          WORD8 = 000020
T$BT  = 000020          T$SEG1= 000000          WORD28= 000070          WORD54= 000154          WORD80= 000240
T$BTAR= 000030          T$SEG2= 000001          WORD29= 000072          WORD55= 000156          WORD81= 000242
T$BTD = 002000          T$SEG3= 000002          WORD3 = 000006          WORD56= 000160          WORD82= 000244
T$CD  = 000100          T$SO  = 000001          WORD30= 000074          WORD57= 000162          WORD83= 000246
T$CLK = 002000          T$UBUS= 100000          WORD31= 000076          WORD58= 000164          WORD84= 000250
T$DISK= 000200          T$1CLK= 000400          WORD32= 000100          WORD59= 000166          WORD85= 000252
T$DRD = 000004          T$8BEN= 000020          WORD33= 000102          WORD6 = 000014          WORD86= 000254
T$EMEM= 010000          UBD.IN= 000020          WORD34= 000104          WORD60= 000170          WORD87= 000256
T$FSAA= 000000          WORD0 = 000000          WORD35= 000106          WORD61= 000172          WORD88= 000260
T$FSAB= 000004          WORD1 = 000002          WORD36= 000110          WORD62= 000174          WORD89= 000262
T$FSAC= 000014          WORD10= 000024          WORD37= 000112          WORD63= 000176          WORD9 = 000022
T$FSB2= 000010          WORD11= 000026          WORD38= 000114          WORD64= 000200          WORD90= 000264
T$IB  = 000026          WORD12= 000030          WORD39= 000116          WORD65= 000202          WORD91= 000266
T$IBAR= 000024          WORD13= 000032          WORD4 = 000010          WORD66= 000204          WORD92= 000270
T$IBE = 020000          WORD14= 000034          WORD40= 000120          WORD67= 000206          WORD93= 000272
T$IBF = 040000          WORD15= 000036          WORD41= 000122          WORD68= 000210          WORD94= 000274
T$ICD = 000040          WORD16= 000040          WORD42= 000124          WORD69= 000212          WORD95= 000276
T$MODE= 004000          WORD17= 000042          WORD43= 000126          WORD7 = 000016          WORD96= 000300
T$OB  = 000036          WORD18= 000044          WORD44= 000130          WORD70= 000214          WORD97= 000302
T$OBE = 004000          WORD19= 000046          WORD45= 000132          WORD71= 000216          WORD98= 000304
T$OBF = 010000          WORD2 = 000004          WORD46= 000134          WORD72= 000220          WORD99= 000306
T$OBRA= 000034          WORD20= 000050          WORD47= 000136          WORD73= 000222          WRDVAL= 000310
T$OBWA= 000032          WORD21= 000052          WORD48= 000140          WORD74= 000224          XTREAD= 001000
T$OUTA= 100000          WORD22= 000054          WORD49= 000142          WORD75= 000226          XTWRTE= 000400

. ABS.  000000     000
        000000     001
CP      001312     002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3201 WORDS  ( 13 PAGES)
DYNAMIC MEMORY:  3860 WORDS  ( 14 PAGES)
ELAPSED TIME:  00:00:42
CP,SP/-SP=[20,1]IM,[20,1]CP
```

```
                                        .TITLE GPBUG1
000000                                  .PSECT CPBUG1
                                ;
                                ;       HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
                                ;       ********   PROTOTYPE VERSION   ********
                                ;       CONTROL PROCESSOR TEST ROUTINES
                                ;
                                ;       COMMANDS:
                                ;       AT      SET BREAKPOINT
                                ;       OF      REMOVE BREAKPOINT
                                ;       LR      LOAD MICROPROCESSOR REGISTERS
                                ;       RG      PRINT MICROPROCESSOR REGISTERS
                                ;       PC      SINGLE STEP A NUMBER OF TIMES
                                ;       SS      SINGLE STEP
                                ;
                                ; **********************
                                ;
                                ;       DISCLAIMER:
                                ;       IN ORDER FOR THE BREAKPOINT ROUTINES TO WORK PROPERLY
                                ;       THERE MUST BE A 'PIPELINE REGISTER ONLY' CLOCK. THIS
                                ;       WOULD ALLOW THE CHANGING OF PLR CONTENTS WITHOUT CAUSING
                                ;       AN INSTRUCTION TO BE EXECUTED. AS IT STANDS NOW, THE PIPE-
                                ;       LINE REGISTER CLOCK IS TIED TO THE SEQUENCER CLOCK. WHEN
                                ;       A BREAKPOINT HAS BEEN REACHED, ANY OPERATION INVOLVING
                                ;       A PLR CLOCK (EG. PRINTING MICROPGM MEMORY) WILL EXECUTE
                                ;       THE INSTRUCTION IN THE PLR. THE 2900 SEQUENCER STACK IS
                                ;       ALSO SOMETIMES CHANGED.
                                ;
                                ; **********************
                                ;
000000  000000                  SSCNT:  .WORD   0               ;NUMBER OF TIMES TO SINGLE STEP
000002  000000                  SSTRT:  .WORD   0               ;WHERE TO START SINGLE STEPPING
                                ;
                                ;       DEBUGGING ROUTINES
                                ;
                                ;       SET BREAKPOINT
                                ;       EG. CP>AT 22
                                ;
000004                          AT2::
000004                                  CALL    FIND            ;LOOK FOR BKPT ADDR IN COMMAND LINE
000010  103004                         BCC     1$              ;OK, CONVERT BKPT ADDRESS
000012                                  CALL    ERR4            ;NOTHING THERE
000016  000167  000000G                 JMP     CPXX
000022                          1$:     CALL    PACK            ;CONVERT BKPT ADDRESS
000026  103004                         BCC     2$              ;OK, CONTINUE
000030                                  CALL    ERR5            ;ERROR ON CONVERSION
000034  000167  000000G                 JMP     CPXX
000040  026767  000000G 000000G 2$:     CMP     CSHIGH,BINWD    ;IS START ADDRESS IN RANGE
000046  003004                         BGT     3$              ;YES, CONTINUE
000050                                  CALL    ERR10           ;ADDR OUT OF RANGE
000054  000167  000000G                 JMP     CPXX
                                ;
000060  012746  000400          3$:     MOV     #<CBKCLKEN>,-(SP)       ;SET CP CNTL FOR BREAKPOINT
000064                                  CALL    CPCR
000070  016746  000000G                 MOV     BINWD,-(SP)             ;LOAD DATA WORD INTO LOAD BUS REG
000074                                  CALL    LBCP            ;SET BCE CNTL WORD
000100  052767  000000G 000000G         BIS     #BREAK,BASE     ;SET FLAG FOR BKPT SET
```

```
                        JMP     CPXX
                ;
                ;
                ;       REMOVE BREAKPOINT (DISABLE)
                ;       CP>OF
                ;
                ;
000112          OF2::
000112  042767  000000G 000000G        BIC     #BREAK,BASE             ;CLEAR BREAKPOINT FLAG
000120  005046                        CLR     -(SP)                   ;SEND 0 TO CP CNTL REG
000122                                CALL    CPCR                    ;EXECUTE TRANSFER
000126  005067  176422                CLR     @R$CR2                  ;CLEAR CSR2
000132  000167  000000G               JMP     CPXX                    ;GET NEXT COMMAND
                ;
                ;
                ;       LOAD ALL CP REGISTERS WITH THE SAME PATTERN
                ;       CP>LR FFFF
                ;
                ;
000136          LR2::
000136                                CALL    FIND                    ;LOOK FOR PATTERN IN COMMAND LINE
000142  103004                        BCC     1$                      ;OK, CONVERT PATTERN
000144                                CALL    ERR4                    ;NOTHING THERE
000150  000167  000000G               JMP     CPXX
000154          1$:                   CALL    PACK                    ;CONVERT PATTERN (SET UP 'BINWD')
000160  103004                        BCC     2$                      ;OK, CONTINUE
000162                                CALL    ERR5                    ;ERROR ON CONVERSION
000166  000167  000000G               JMP     CPXX
                ;
                ;       CALL SUBROUTINE IN CP TO LOAD REGISTERS
                ;
000172  012704  000020  2$:           MOV     #16.,R4                 ;16 REGISTERS
000176  005000                        CLR     R0                      ;REGISTER INDICATOR
000200          3$:                   CALL    LDREG                   ;LOAD A CP REGISTER
000204  005200                        INC     R0                      ;INC TO NEXT REG
000206  005304                        DEC     R4                      ;FINISHED ALL REGS ?
000210  001373                        BNE     3$                      ;NO, CONTINUE
000212  000167  000000G               JMP     CPXX
                ;
                ;
                ;       PRINT CP REGISTERS
                ;       CP>RG
                ;       CALL SUBROUTINE IN CP TO READ REGISTERS
                ;
                ;
000216          RG2::
000216  012704  000002                MOV     #2,R4                   ;PRINT IN 2 SETS OF 8 REGS EACH
000222  005000                        CLR     R0                      ;REGISTER INDICATOR
000224  012705  000000G       1$:     MOV     #PRINT,R5               ;R5 -> PRINT LINE
000230  012703  000010                MOV     #8.,R3                  ;PRINT 8 REGS ON A LINE
000234          2$:                   CALL    REREG                   ;READ A CP REGISTER
000240                                CALL    UNPK                    ;CONVERT TO ASCII
000244  005205                        INC     R5                      ;BUMP PRINT LINE POINTER
000246  005200                        INC     R0                      ;BUMP REG INDICATOR
000250  005303                        DEC     R3
000252  001370                        BNE     2$                      ;STAY ON CURRENT LINE
000254                                CALL    CONSOL                  ;WRITE LINE TO CONSOLE
```

```
000260  005304                          DEC     R4
000262  001360                          BNE     1$                      ;DO SECOND SET OF 8
000264  000167  000000G                 JMP     CPXX
                                ;
                                ;
                                ;       SINGLE STEP A NUMBER OF TIMES
                                ;       CP>PC 4
                                ;
                                ;
000270                          PC2::
000270                                  CALL    FIND                    ;LOOK FOR COUNT IN COMMAND LINE
000274  103004                          BCC     1$                      ;OK, CONTINUE
000276                                  CALL    ERR4                    ;COUNT MISSING
000302  000167  000000G                 JMP     CPXX
000306                          1$:     CALL    PACK                    ;CONVERT COUNT
000312  103004                          BCC     2$                      ;OK, CONTINUE
000314                                  CALL    ERR5                    ;ERROR ON CONVERSION
000320  000167  000000G                 JMP     CPXX
                                ;
000324  005767  000000G         2$:     TST     BINWD                   ;COUNT = 0 ?
000330  001004                          BNE     3$                      ;NO, CONTINUE
000332                                  CALL    ERR5                    ;COUNT MUST BE GT 0
000336  000167  000000G                 JMP     CPXX
000342  016767  000000G 177430  3$:     MOV     BINWD,SSCNT             ;SAVE COUNT
000350  000460                          BR      SSCLR                   ;ENTER SS COMMAND
                                ;
                                ;
                                ;       SINGLE STEP
                                ;       CP>SS
                                ;       CP>SS 6
                                ;       CP>SS 6 2
                                ;
                                ;
000352                          SS2::
000352                                  CALL    FIND                    ;LOOK FOR START ADDR IN COMMAND LINE
000356  103455                          BCS     SSCLR                   ;NONE THERE, USE CURRENT MAR
000360                                  CALL    PACK                    ;CONVERT START ADDRESS
000364  103004                          BCC     1$                      ;OK, CONTINUE
000366                                  CALL    ERR5                    ;ERROR ON CONVERSION
000372  000167  000000G                 JMP     CPXX
                                ;
000376  026767  000000G 000000G 1$:     CMP     CSHIGH,BINWD            ;IS START ADDRESS IN RANGE
000404  003004                          BGT     2$                      ;YES, CONTINUE
000406                                  CALL    ERR10                   ;OUT OF RANGE HIGH
000412  000167  000000G                 JMP     CPXX
                                ;
                                ;       SS START ADDRESS FOUND AND CONVERTED. NOW LOOK FOR
                                ;       COUNT (NUMBER OF SINGLE STEPS). EG. IN THE COMMAND:
                                ;       CP>SS 6 2
                                ;       LOOK FOR THE '2'
                                ;
000416  016767  000000G 177356  2$:     MOV     BINWD,SSTRT             ;SAVE START ADDRESS
000424                                  CALL    FIND                    ;LOOK FOR COUNT IN COMMAND LINE
000430  103421                          BCS     5$                      ;NOTHING THERE, USE DEFAULT
000432                                  CALL    PACK                    ;CONVERT COUNT
000436  103004                          BCC     3$                      ;OK, CONTINUE
000440                                  CALL    ERR5                    ;ERROR ON CONVERSION
```

```
172 000444  000167  000000G                 JMP     CPXX
173                                 ;
174 000450  005767  000000G         3$:     TST     BINWD                   ;COUNT = 0 ?
175 000454  001004                          BNE     4$                      ;NO, CONTINUE
176 000456                                   CALL    ERR5                    ;COUNT MUST BE GT 0
177 000462  000167  000000G                 JMP     CPXX
178 000466  016767  000000G 177304  4$:     MOV     BINWD,SSCNT             ;SAVE COUNT
179                                 ;
180                                 ;       SEQUENCE UP TO SINGLE STEP ADDRESS (IF NECESSARY)
181                                 ;       SINGLE CLOCK ALL TO EXECUTE ONE INSTRUCTION
182                                 ;       NB. SINGLE STEP CANNOT BE USED IN SENDING THE
183                                 ;       BCL ADDRESS TO THE CP.
184                                 ;
185 000474  016746  177302          5$:     MOV     SSTRT,-(SP)             ;PROVIDE START ADDRESS FOR SEQUENCER
186 000500                                   CALL    SEQCS                   ;SEQUENCE ONLY UP TO START ADDRESS
187 000504  005046                           CLR     -(SP)                   ;CLEAR CP CONTROL REG
188 000506                                   CALL    CPCR
189                                 ;
190                                 ;       SET UP FOR SINGLE CLOCK
191                                 ;
192 000512  012767  100000 176422  SSCLR:  MOV     #Q$SM,QR$CR2            ;SET SEARCH MODE
193 000520  012746  036000          3$:     MOV     #<Q$CNC+Q$RNC>,-(SP)    ;CLEAR PPS AND CP NO-CLOCKS
194 000524  052716  000360                  BIS     #<Q$CSEL>,(SP)          ;CLEAR SELECT BITS
195 000530  052716  001001                  BIS     #<Q$LBD+Q$LBP>,(SP)     ;CLEAR DRIVE AND PULSE
196 000534  012746  020000                  MOV     #Q$CSP,-(SP)            ;SET SINGLE CLOCK ALL
197 000540                                   CALL    CSR1                    ;WRITE CSR1
198 000544  005046                           CLR     -(SP)                   ;CLEAR NOTHING
199 000546  012746  036000                  MOV     #<Q$CNC+Q$RNC>,-(SP)    ;SET PPS AND CP NO-CLOCKS
200 000552                                   CALL    CSR1                    ;AND RE-WRITE CSR1
201                                 ;
202                                 ;       SINGLE STEP A NUMBER OF TIMES DEPENDING UPON THE
203                                 ;       COUNT IN FIELD 'SSCNT'.
204                                 ;
205 000556  005067  176422                  CLR     QR$CR2                  ;SET LOAD MODE
206 000562  005767  177212                  TST     SSCNT                   ;SS A NUMBER OF TIMES ?
207 000566  001403                          BEQ     4$                      ;NO, EXIT SS RTN
208 000570  005367  177204                  DEC     SSCNT
209 000574  001346                          BNE     SSCLR                   ;EXECUTE NEXT INSTRUCTION
210                                 ;
211 000576                          4$:     CALL    CPMP                    ;PRINT MICROPGM ADDRESS
212 000602  000167  000000G                 JMP     CPXX
```

```
                                        ;
                                        ;
                                        ;       PRINT CP MICROPROGRAM ADDRESS
                                        ;
                                        ;
000606                                  CPMP:
000606  012746  000004                          MOV     #<CSADRD>,-(SP)         ;READ CS ADDRESS SELECT BIT
000612                                          CALL    CPCR                    ;WRITE CP CONTROL REGISTER
000616                                          CALL    CPLB                    ;REQUEST CP TO LOD BUS
000622  012601                                  MOV     (SP)+,R1                ;UNLOAD MPA
000624  042701  176000                          BIC     #176000,R1              ;CLEAR UPPER 6 BITS
000630  012705  000000G                         MOV     #PRINT,R5               ;POINT TO PRINT LINE
000634                                          CALL    UNPK                    ;CONVERT MPA
000640                                          CALL    CONSOL                  ;WRITE TO CONSOLE
000644  005046                                  CLR     -(SP)                   ;CLEAR CP CR
000646                                          CALL    CPCR
000652                                          RETURN
                                        ;
000001                                          .END
```

```
ALUCKE= 040000E          BYTE39= 000047          BYTE90= 000132          MMOE  = 000004          Q$MSET= 000004
ALUOE = 004000           BYTE4 = 000004          BYTE91= 000133          MMWRTE= 000010          Q$MSP = 100000
AT2     000004RG   002   BYTE40= 000050          BYTE92= 000134          MNOBRE= 100000          Q$NCLK= 176000
A01   = 010000           BYTE41= 000051          BYTE93= 000135          MREN1 = 000001          Q$PP  = 000100
BASE  = ****** GX        BYTE42= 000052          BYTE94= 000136          MREN2 = 020000          Q$PPSW= 000320
BINWD = ****** GX        BYTE43= 000053          BYTE95= 000137          MSYN  = 000040          Q$PP2 = 000300
BITVAL= 000000           BYTE44= 000054          BYTE96= 000140          N     = 000144          Q$QHLT= 000013
BIT0  = 000001           BYTE45= 000055          BYTE97= 000141          OF2     000112RG  002   Q$QL  = 000043
BIT1  = 000002           BYTE46= 000056          BYTE98= 000142          PACK  = ****** GX       Q$QLA = 000053
BIT10 = 002000           BYTE47= 000057          BYTE99= 000143          PC2     000270RG  002   Q$QLB = 000054
BIT11 = 004000           BYTE48= 000060          BYTVAL= 000144          PLB   = 000010          Q$QLR = 000001
BIT12 = 010000           BYTE49= 000061          CB.ALL= 001000          PLC   = 000020          Q$QW  = 000042
BIT13 = 020000           BYTE5 = 000005          CBKCLK= 000400          PLD   = 000030          Q$RDCD= 000005
BIT14 = 040000           BYTE50= 000062          CNOBRE= 100000          PLRWR = 000200          Q$RDMD= 000006
BIT15 = 100000           BYTE51= 000063          CONSOL= ****** GX       PLR.EN= 000200          Q$REBK= 001000
BIT2  = 000004           BYTE52= 000064          CPCCEN= 010000          PRINT = ****** GX       Q$RNC = 006000
BIT3  = 000010           BYTE53= 000065          CPCR  = ****** GX       QR$CR1= 176420          Q$RSC = 004000
BIT4  = 000020           BYTE54= 000066          CPLB  = ****** GX       QR$CR2= 176422          Q$RSET= 000010
BIT5  = 000040           BYTE55= 000067          CPMP    000606R   002   QR$LBR= 176424          Q$SM  = 100000
BIT6  = 000100           BYTE56= 000070          CPREAD= 040000          Q$ATTN= 000100          Q$SP  = 000120
BIT7  = 000200           BYTE57= 000071          CPWRTE= 020000          Q$BCL = 000001          Q$SP2 = 000340
BIT8  = 000400           BYTE58= 000072          CPXX  = ****** GX       Q$CCCP= 000040          REREG = ****** GX
BIT9  = 001000           BYTE59= 000073          CSADRD= 000004          Q$CHB = 000400          RGQ.EN= 000200
BREAK = ****** GX        BYTE6 = 000006          CSEQCI= 100000          Q$CHRL= 000200          RGQ.VA= 020000
BYTE0 = 000000           BYTE60= 000074          CSHIGH= ****** GX       Q$CLR = 000040          RG2     000216RG  002
BYTE1 = 000001           BYTE61= 000075          CSOE  = 000040          Q$CNC = 030000          SEQCS = ****** GX
BYTE10= 000012           BYTE62= 000076          CSR1  = ****** GX       Q$CP  = 000060          SEQ.CI= 000010
BYTE11= 000013           BYTE63= 000077          CSWRTE= 000100          Q$CPCC= 000010          SSCLR   000512R   002
BYTE12= 000014           BYTE64= 000100          DBR.RD= 000001          Q$CP2 = 000260          SSCNT   000000R   002
BYTE13= 000015           BYTE65= 000101          DB$CPP= 001457          Q$CSC = 010000          SSTRT   000002R   002
BYTE14= 000016           BYTE66= 000102          DB$SPT= 000026          Q$CSEL= 000360          SS2     000352RG  002
BYTE15= 000017           BYTE67= 000103          DB$TPC= 000023          Q$CSET= 000002          S$CLR = 000000
BYTE16= 000020           BYTE68= 000104          DISPGS= 100000          Q$CSP = 020000          S$LA  = 000001
BYTE17= 000021           BYTE69= 000105          DMAAWR= 000005          Q$DMA = 000001          S$QB  = 000005
BYTE18= 000022           BYTE7 = 000007          DMARRD= 000003          Q$ENBK= 040000          S$QR  = 000006
BYTE19= 000023           BYTE70= 000106          DMARWR= 000004          Q$ENOP= 020000          S$QX  = 000004
BYTE2 = 000002           BYTE71= 000107          ENBR  = 010000          Q$FAL = 004000          S$SR  = 000007
BYTE20= 000024           BYTE72= 000110          ERR10 = ****** GX       Q$FC  = 000045          S$S1  = 000010
BYTE21= 000025           BYTE73= 000111          ERR4  = ****** GX       Q$FD  = 000044          S$S2  = 000014
BYTE22= 000026           BYTE74= 000112          ERR5  = ****** GX       Q$FP  = 000046          TD$CTR= 176370
BYTE23= 000027           BYTE75= 000113          FIND  = ****** GX       Q$HBF = 000002          TD$CTW= 176360
BYTE24= 000030           BYTE76= 000114          LBCP  = ****** GX       Q$ICP = 000006          TD$INL= 004000
BYTE25= 000031           BYTE77= 000115          LDREG = ****** GX       Q$IHB = 000003          TD$MEM= 000270
BYTE26= 000032           BYTE78= 000116          LOC.EN= 000100          Q$IHRL= 000002          TD$OAR= 176344
BYTE27= 000033           BYTE79= 000117          LOC.WA= 040000          Q$IMRP= 000007          TD$OTR= 176346
BYTE28= 000034           BYTE8 = 000010          LOC.WB= 100000          Q$LBD = 001000          TD$ORD= 000274
BYTE29= 000035           BYTE80= 000120          LR2     000136RG  002   Q$LBDP= 001001          TD$SW = 176376
BYTE3 = 000003           BYTE81= 000121          MAREN1= 000001          Q$LBP = 000001          TD$TAR= 176372
BYTE30= 000036           BYTE82= 000122          MAREN2= 004000          Q$LDCD= 000003          TD$TAW= 176362
BYTE31= 000037           BYTE83= 000123          MARLOD= 010000          Q$LDMD= 000004          TD$TDR= 176374
BYTE32= 000040           BYTE84= 000124          MAROUT= 000002          Q$LDPP= 002000          TD$TDW= 176364
BYTE33= 000041           BYTE85= 000125          MAR.LO= 002000          Q$LHP = 010000          T$AD  = 000020
BYTE34= 000042           BYTE86= 000126          MAR.OU= 000040          Q$MNC = 140000          T$BA  = 000002
BYTE35= 000043           BYTE87= 000127          MBKALL= 001000          Q$MR  = 000052          T$BD  = 000010
BYTE36= 000044           BYTE88= 000130          MBKCLK= 000400          Q$MRP = 000040          T$BSO = 100000
BYTE37= 000045           BYTE89= 000131          MMADRD= 000100          Q$MRP2= 000240          T$BT  = 000020
BYTE38= 000046           BYTE9 = 000011          MMLEFT= 000002          Q$MSC = 040000          T$BTAR= 000030
```

SYMBOL TABLE

```
T$BTD = 002000        T$SEG2= 000001        WORD27= 000066        WORD52= 000150        WORD78= 000234
T$CD  = 000100        T$SEG3= 000002        WORD28= 000070        WORD53= 000152        WORD79= 000236
T$CLK = 002000        T$SO  = 000001        WORD29= 000072        WORD54= 000154        WORD8 = 000020
T$DISK= 000200        T$UBUS= 100000        WORD3 = 000006        WORD55= 000156        WORD80= 000240
T$DRD = 000004        T$1CLK= 000400        WORD30= 000074        WORD56= 000160        WORD81= 000242
T$EMEM= 010000        T$8BEN= 000020        WORD31= 000076        WORD57= 000162        WORD82= 000244
T$FSAA= 000000        UBD.IN= 000020        WORD32= 000100        WORD58= 000164        WORD83= 000246
T$FSAB= 000004        UNPK  = ****** GX     WORD33= 000102        WORD59= 000166        WORD84= 000250
T$FSAC= 000014        WORD0 = 000000        WORD34= 000104        WORD6 = 000014        WORD85= 000252
T$FSB2= 000010        WORD1 = 000002        WORD35= 000106        WORD60= 000170        WORD86= 000254
T$IB  = 000026        WORD10= 000024        WORD36= 000110        WORD61= 000172        WORD87= 000256
T$IBAR= 000024        WORD11= 000026        WORD37= 000112        WORD62= 000174        WORD88= 000260
T$IBE = 020000        WORD12= 000030        WORD38= 000114        WORD63= 000176        WORD89= 000262
T$IBF = 040000        WORD13= 000032        WORD39= 000116        WORD64= 000200        WORD9 = 000022
T$ICD = 000040        WORD14= 000034        WORD4 = 000010        WORD65= 000202        WORD90= 000264
T$MODE= 004000        WORD15= 000036        WORD40= 000120        WORD66= 000204        WORD91= 000266
T$OB  = 000036        WORD16= 000040        WORD41= 000122        WORD67= 000206        WORD92= 000270
T$OBE = 004000        WORD17= 000042        WORD42= 000124        WORD68= 000210        WORD93= 000272
T$OBF = 010000        WORD18= 000044        WORD43= 000126        WORD69= 000212        WORD94= 000274
T$OBRA= 000034        WORD19= 000046        WORD44= 000130        WORD7 = 000016        WORD95= 000276
T$OBWA= 000032        WORD2 = 000004        WORD45= 000132        WORD70= 000214        WORD96= 000300
T$OUTA= 100000        WORD20= 000050        WORD46= 000134        WORD71= 000216        WORD97= 000302
T$RBDO= 000200        WORD21= 000052        WORD47= 000136        WORD72= 000220        WORD98= 000304
T$RNB = 000040        WORD22= 000054        WORD48= 000140        WORD73= 000222        WORD99= 000306
T$RSET= 040000        WORD23= 000056        WORD49= 000142        WORD74= 000224        WRDVAL= 000310
T$SC  = 000022        WORD24= 000060        WORD5 = 000012        WORD75= 000226        XTREAD= 001000
T$SCLK= 020000        WORD25= 000062        WORD50= 000144        WORD76= 000230        XTWRTE= 000400
T$SEG1= 000000        WORD26= 000064        WORD51= 000146        WORD77= 000232
```

```
. ABS.  000000     000
        000000     001
CPBUG1  000654     002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3220 WORDS  ( 13 PAGES)
DYNAMIC MEMORY:  3860 WORDS  ( 14 PAGES)
ELAPSED TIME:  00:00:44
LPBUG1,LPBUG1/-SP=[20,1]IM,[20,1]CPBUG1
```

```
 1                                      .TITLE.CPBUG2.
 2.000000                               .PSECT.CPBUG2.
 3                                ;
 4                                ;
 5                                ;      HARDWARE.QUERY.RESOLVER.'MANUAL' DEBUGGING AIDS.
 6                                ;      CONTROL.PROCESSOR.TEST ROUTINES.
 7                                ;      *******   PROTOTYPE.VERSION.   *******
 8                                ;
 9                                ;      COMMANDS:
10                                ;      GO       PROCEED.FROM.BREAKPOINT.
11                                ;      FR       FREE RUN.
12.                               ;
13                                ;
14                                ; ********************
15                                ;
16                                ;      DISCLAIMER:
17                                ;      IN ORDER.FOR THE.BREAKPOINT.ROUTINES.TO.WORK.PROPERLY.
18                                ;      THERE.MUST.BE.A.'PIPELINE.REGISTER.ONLY' CLOCK. THIS
19                                ;      WOULD.ALLOW.THE.CHANGING.OF.PLR.CONTENTS.WITHOUT.CAUSING.
20                                ;      AN INSTRUCTION.TO.BE.EXECUTED, AS.IT.STANDS.NOW, THE PIPE-
21                                ;      LINE.REGISTER.CLOCK.IS TIED.TO.THE.SEQUENCER.CLOCK. WHEN.
22.                               ;      A.BREAKPOINT HAS.BEEN.REACHED, ANY.OPERATION.INVOLVING.
23                                ;      A.PLR.CLOCK.(EG. PRINTING.MICROPGM.MEMORY) WILL.EXECUTE.
24                                ;      THE.INSTRUCTION.IN.THE PLR. THE.2900 SEQUENCER.STACK IS
25                                ;      ALSO.SOMETIMES.CHANGED.
26                                ;
27                                ; ********************
28.                               ;
29                                ;
30                                      .MCALL.  WTSE$S,CLEF$S,WTLO$S,RDAF$S.
31                               ;
32.                              ;
33                                      .NLIST.  BEX.
34 000000   115   122   120  MRPMSG: .ASCIZ.  /MRP BREAKPOINT/.
35 000017   110   111   124  HBMSG:  .ASCIZ.  /HIT LIST.INTERRUPT/.
36 000042.  110   122   114  HRLMSG: .ASCIZ.  /HRL INTERRUPT/.
37 000060   104   115   101  DMAMSG: .ASCIZ.  /DMA INTERRUPT/.
38 000076   125   116   111  UNMSG:  .ASCII.  /UNIDENTIFIED.INTERRUPT. CSR2.=./.
39                                      .EVEN.
40                                      .NLIST.  CND.
```

```
                                ;
                                ;
                                ;       DEBUGGING ROUTINES
                                ;
                                ;
                                ;       PROCEED
                                ;       THE 'GO' COMMAND RUNS THE CP MICROCODE ONLY
                                ;       TO RUN BOTH CP AND MRP USE 'FR'
                                ;       EG. CP>GO
                                ;           CP>GO 22
                                ;
000136                          GO2::
000136                                  CALL    FIND                    ;LOOK FOR START ADDRESS
000142  103443                         BCS     CGO                     ;NONE, GO FROM PRESENT ADDRESS
000144                                  CALL    PACK                    ;CONVERT START ADDRESS
000150  103004                         BCC     1$                      ;OK, CONTINUE
000152                                  CALL    ERR5                    ;ERROR ON CONVERSION
000156  000167  000000G                JMP     CPXX
                                ;
000162  026767  000000G 000000G 1$:     CMP     CSHIGH,BINWD            ;IS START ADDRESS IN RANGE
000170  003004                         BGT     2$                      ;YES, CONTINUE
000172                                  CALL    ERR10                   ;OUT OF RANGE HIGH
000176  000167  000000G                JMP     CPXX
                                ;
                                ;       IF THE COMMAND IS:
                                ;       CP>GO 0
                                ;       THEN SCAN THE COMMAND LINE FURTHER FOR AN 'N'
                                ;       CP>GO 0 N
                                ;       THE 'N' MEANS NOT TO ERASE THE QLB'S SO THAT THEY MAY
                                ;       BE LOOKED AT.
                                ;
000202  005767  000000G         2$:     TST     BINWD                   ;GO FROM ZERO
000206  001015                         BNE     4$                      ;NO, CONTINUE
000210                                  CALL    FIND                    ;LOOK FOR 'N' IN COMMAND LINE
000214  103412                         BCS     4$                      ;NOT THERE, ERASE QLB
000216  122711  000116                 CMPB    #'N,(R1)                ;MAKE SURE IT IS AN 'N'
000222  001404                         BEQ     3$                      ;OK, CONTINUE
000224                                  CALL    ERR12
000230  000167  000000G                JMP     CPXX                    ;BAD COMMAND
000234  052767  000000G 000000G 3$:     BIS     #NEQLB,BASE             ;SET FLAG FOR NO QLB ERASE
000242  016746  000000G         4$:     MOV     BINWD,-(SP)             ;PROVIDE START ADDRESS FOR SEQUENCER
000246                                  CALL    SEQCS                   ;SEQUENCE ONLY UP TO START ADDRESS
                                ;
                                ;       SET UP TO FIELD INTERRUPTS
                                ;       'OPERATIONAL INTERRUPTS' ARE HIT BUFFER INTERRUPTS, HRL
                                ;       INTERRUPTS, AND DMA INTERRUPTS.
                                ;
                                CGO:
000252                                  MOV     #<Q$CHB+Q$CHRL>,QR$CR2  ;CLEAR OPERATIONAL INTERRUPTS
000252  012767  000600  176422          MOV     #Q$REBK,QR$CR2          ;RE-ARM INTERRUPTS
000260  012767  001000  176422
                                ;
                                ;       READ MICROPGM ADDRESS TO SEE WHERE SEQUENCER
                                ;       IS CURRENTLY SITTING
                                ;
000266  012746  000004                  MOV     #CSADRD,-(SP)
000272                                  CALL    CPCR                    ;ENABLE CS ADDRESS - ALSO REINHIBIT BR
000276                                  CALL    CPLB                    ;REQUEST CP TO LOD BUS
```

```
 99 000302  005046                          CLR     -(SP)                   ; CLEAR THE CONTROL REGISTER    NS
100 000304                                   CALL    CPCR                    ;                               NS
101 000310  012601                           MOV     (SP)+,R1                ;CP WORD RETURNED ON STACK
102 000312  042701  176000                   BIC     #176000,R1              ;CLEAR UNUSED BITS
103                                  ;
104                                  ;       IF GO FROM ZERO, INITIALIZE PPS AND CLEAR HIT BUFFER
105                                  ;
106 000316  005701                           TST     R1                      ;START FROM ZERO
107 000320  001011                           BNE     10$                     ;NO, CONTINUE
108 000322                                   CALL    PPINIT                  ;DO IT
109                                  ;
110 000326  012700  000000G                  MOV     #HLB,R0                 ;POINT TO HIT BUFFER
111 000332  016702  000000G                  MOV     BCL,R2                  ;LOAD NUMBER OF WORDS IN HIT BUFFER
112 000336  005020                   20$:    CLR     (R0)+                   ;CLEAR HIT LIST
113 000340  005302                           DEC     R2
114 000342  001375                           BNE     20$
115                                  ;
116                                  ;       IF A BREAKPOINT IS IN EFFECT, BRANCH TO CODE THAT PREPARES
117                                  ;       TO FIELD A BREAKPOINT INTERRUPT, ELSE START FREE RUN
118                                  ;       WITHOUT BREAKPOINTS.
119                                  ;
120 000344  032767  000000G 000000G 10$:    BIT     #BREAK,BASE             ;PROCEED FROM BREAKPOINT
121 000352  001075                           BNE     CBK                     ;YES, SET UP FOR BREAKPOINT
122 000354  012767  120000  176422           MOV     #<Q$SM+Q$ENOP>,QR$CR2   ;ENABLE OPERATIONAL INTERRUPTS
123 000362  012746  000360                   MOV     #<Q$CSEL>,-(SP)         ;CLR SELECT BITS
124 000366  052716  001001                   BIS     #<Q$LBD+Q$LBP>,(SP)     ;CLEAR DRIVE AND PULSE
125 000372  052716  036000                   BIS     #<Q$CNC+Q$RNC>,(SP)     ;CLEAR PPS AND CP NO-CLOCKS
126 000376  005046                           CLR     -(SP)                   ;SET NOTHING
127 000400                                   CALL    CSR1                    ;RE-WRITE CSR1
128                                  ;
129                                  ;       IF STARTING FROM ZERO, SEND THE BCL ADDRESS OVER TO
130                                  ;       THE CP.
131                                  ;
132 000404  005701                           TST     R1                      ;START FROM ZERO ?
133 000406  001026                           BNE     1$                      ;WORD NOT ZERO
134 000410  012767  000001  176424           MOV     #Q$BCL,QR$LBR           ;SET ATTN CODE = 1
135 000416  012767  120100  176422           MOV     #<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2 ;SET ATTN CODE READY
136 000424  016701  176422           11$:    MOV     QR$CR2,R1               ;READ CSR2
137 000430  032701  000100                   BIT     #Q$ATTN,R1              ;IS ATTN BIT CLEAR
138 000434  001373                           BNE     11$                     ;NO, LOOP UNTIL IT IS
139                                  ;
140 000436  012767  000000G 176424           MOV     #BCL,QR$LBR             ;MOVE BCL ADDR TO LOD BUS REG
141 000444  012767  120040  176422           MOV     #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP SLAVE DATA READY
142 000452  016701  176422           12$:    MOV     QR$CR2,R1               ;READ CSR2
143 000456  032701  000040                   BIT     #Q$CCCP,R1              ;IS CC TO CP CLEAR
144 000462  001373                           BNE     12$                     ;NO, LOOP UNTIL IT IS
145                                  ;
146                                  ;       PROVIDE FOR INTERRUPT FROM TERMINAL TO RETURN CONTROL
147                                  ;       TO USER (IN CASE MICROCODE HANGS UP OR RUNS FOREVER),
148                                  ;       CALL SUBROUTINE THAT WAITS FOR INTERRUPTS THEN DECODES
149                                  ;       THEM.
150                                  ;
151 000464                           1$:     CLEF$S  #EFN.4                  ;CLEAR QIO EVENT FLAG
152                                  ;
153 000476                                   CALL    HANG2                   ;ISSUE QIO TO TERMINAL
154 000502                                   CALL    WINT                    ;WAIT FOR EVENTS
155 000506                                   CALL    KILL                    ;KILL QIO
```

```
                                        ;
                                        ;       TURN OFF CLOCKS, SET LOAD MODE, PRINT MICROPGM ADDRESS.
                                        ;
000512  005046                          CLR     -(SP)                   ;CLEAR NOTHING.
000514  012746  036000                  MOV     #<Q$CNC+Q$RNC>,-(SP)    ;SET PPS AND CP NO-CLOCKS.
000520                                  CALL    CSR1                    ;WRITE CSR 1
000524  005046                          CLR     -(SP)                   ;SET CP CONTROL REG TO ZERO.
000526                                  CALL    CPCR                    ;DO IT.
000532  005067  176422                  CLR     QR$CR2.                 ;SET LOAD MODE.
000536                                  CALL    CPMP                    ;PRINT MICROPGM ADDRESS.
000542  000167  000000G                 JMP     CPXX
                                        ;
                                        ;
                                        ;       EXPECT TO HIT A BREAKPOINT.
                                        ;
000546                                  CBK:
000546                                  CLEF$S  #EFN.3                  ;CLEAR BREAKPOINT EVENT FLAG.
                                        ;
000560                                  CLEF$S  #EFN.4                  ;CLEAR QIO EVENT FLAG.
                                        ;
                                        ;       SET BKPT ALLOW BIT IN CP CONTROL REG.
                                        ;       SET SEARCH MODE.
                                        ;       SET FREE RUN CLOCK.
                                        ;
000572  012746  001000                  MOV     #CBKALLOW,-(SP)         ;SET CP CONTROL BIT.
000576                                  CALL    CPCR                    ;WRITE CP CONTROL WORD.
000602  012767  160000  176422          MOV     #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2    ;SET SEARCH MODE.
000610  012746  000360                  MOV     #<Q$CSEL>,-(SP)         ;CLR SELECT BITS.
000614  052716  001001                  BIS     #<Q$LBD+Q$LBP>,(SP)     ;CLEAR DRIVE AND PULSE.
000620  052716  036000                  BIS     #<Q$CNC+Q$RNC>,(SP)     ;CLEAR PPS AND CP NO-CLOCKS.
000624  005046                          CLR     -(SP)                   ;SET NOTHING.
000626                                  CALL    CSR1                    ;RE-WRITE CSR1
                                        ;
                                        ;       IF STARTING FROM ZERO, SEND THE BCL ADDRESS OVER TO
                                        ;       THE CP.
                                        ;
000632  005701                          TST     R1                      ;START FROM ZERO ?
000634  001026                          BNE     1$                      ;WORD NOT ZERO.
000636  012767  000001  176424          MOV     #Q$BCL,QR$LBR           ;SET ATTN CODE = 1
000644  012767  120100  176422          MOV     #<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2 ;SET ATTN CODE READY
000652  016701  176422          11$:    MOV     QR$CR2,R1               ;READ CSR2
000656  032701  000100                  BIT     #Q$ATTN,R1              ;IS ATTN BIT CLEAR.
000662  001373                          BNE     11$                     ;NO, LOOP UNTIL IT IS.
                                        ;
000664  012767  000000G 176424          MOV     #BCL,QR$LBR             ;MOVE BCL ADDR TO LOD BUS REG.
000672  012767  120040  176422          MOV     #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP SLAVE DATA READY.
000700  016701  176422          12$:    MOV     QR$CR2,R1               ;READ CSR2
000704  032701  000040                  BIT     #Q$CCCP,R1              ;IS CC TO CP CLEAR.
000710  001373                          BNE     12$                     ;NO, LOOP UNTIL IT IS.
                                        ;
                                        ;       PROVIDE FOR INTERRUPT FROM TERMINAL TO RETURN CONTROL
                                        ;       TO USER (IN CASE MICROCODE HANGS UP OR RUNS FOREVER).
                                        ;       CALL SUBROUTINE THAT WAITS FOR INTERRUPTS THEN DECODES
                                        ;       THEM.
                                        ;
000712                          1$:     CALL    HANG2                   ;ISSUE QIO
000716                                  CALL    WINT                    ;WAIT FOR INTERRUPTS
```

```
                                        ;
                                        ;       RESET·NO-CLOCKS·IN·CSR 1
                                        ;       SET·LOAD·MODE·
                                        ;       PRINT·MICROPGM·ADDRESS
                                        ;
000722· 005046                          CLR·    -(SP)                   ;CLEAR·NOTHING·
000724  012746  036000                  MOV·    #<Q$CNC+Q$RNC>,-(SP)    ;SET·PPS·AND·CP·NO-CLOCKS·
000730                                  CALL·   CSR1                    ;RE-WRITE·CSR1
000734  005046                          CLR·    -(SP)                   ;SEND·0 TO CP·CONTROL·REG·
000736                                  CALL·   CPCR                    ;DO·IT·
                                        ;
000742· 005067  176422                  CLR·    QR$CR2·                 ;SET·LOAD MODE·
000746  012767  001000  176422·         MOV·    #Q$REBK,QR$CR2          ;RE-ARM· INTERRUPTS·
000754                                  CALL·   CPMP                    ;PRINT·MICROPGM·ADDRESS·
000760                                  CALL·   KILL                    ;KILL·AST·
000764  000167  000000G·                JMP·    CPXX
```

```
                                        ;
                                        ;
                                        ;       FREE RUN
                                        ;       RUN BOTH MRP AND CP
                                        ;       EG  CP>FR
                                        ;           CP>FR N
                                        ;       THE 'N' MEANS NOT TO ERASE THE QLB'S SO THAT THEY MAY
                                        ;       BE LOOKED AT.
                                        ;
                                        ;
000770                                  FR2::
000770                                          CALL    FIND                    ;LOOK FOR 'N' IN COMMAND LINE
000774  103412                                  BCS     20$                     ;NOT THERE, ERASE QLB
000776  122711  000116                          CMPB    #'N,(R1)                ;MAKE SURE IT IS AN 'N'
001002  001404                                  BEQ     10$                     ;OK, CONTINUE
001004                                          CALL    ERR12
001010  000167  000000G                         JMP     CPXX                    ;BAD COMMAND
001014  052767  000000G 000000G       10$:     BIS     #NEQLB,BASE             ;SET FLAG FOR NO QLB ERASE
                                        ;
                                        ;       LOAD PPS MASK REGISTER
                                        ;       DO A MASTER RESET
                                        ;       INITIALIZE PPS
                                        ;
001022  012746  000052                  20$:    MOV     #Q$MR,-(SP)             ;SELECT PPS MASK REGISTER
001026                                          CALL    PPCR
001032  012746  037774                          MOV     #037774,-(SP)           ;WRITE X'3FFC'
001036                                          CALL    LBPSC
                                        ;
001042  012746  177777                          MOV     #177777,-(SP)           ;CLEAR CSR1
001046  012746  000010                          MOV     #Q$RSET,-(SP)           ;SET RESET
001052                                          CALL    CSR1                    ;RESET HQR
001056  012746  000010                          MOV     #Q$RSET,-(SP)           ;CLEAR RESET
001062  012746  176000                          MOV     #<Q$MNC+Q$CNC+Q$RNC>,-(SP);SET NO CLKS
001066                                          CALL    CSR1                    ;MOVE TO CSR1
                                        ;
001072                                          CALL    PPINIT                  ;INITIALIZE PPS
                                        ;
                                        ;       SET MRP AND CP START ADDRESSES TO ZERO
                                        ;
001076  005046                                  CLR     -(SP)                   ;START MRP AT ZERO
001100                                          CALL    SEQMM                   ;SET START ADDRESS
                                        ;
001104  005046                                  CLR     -(SP)                   ;REINHIBIT BRANCH CONTROL REGISTER
001106                                          CALL    MRPCR
                                        ;
001112  005046                                  CLR     -(SP)                   ;START CP AT ZERO
001114                                          CALL    SEQCS                   ;SEQUENCE ONLY UP TO START ADDRESS
                                        ;
001120  005046                                  CLR     -(SP)                   ;REINHIBIT BRANCH CONTROL REGISTER
001122                                          CALL    CPCR
                                        ;
                                        ;       SET UP FOR OPERATIONAL INTERRUPTS,
                                        ;       'OPERATIONAL INTERRUPTS' ARE HIT BUFFER INTERRUPTS,
                                        ;       HRL INTERRUPTS, AND DMA INTERRUPTS.
                                        ;       CLEAR HIT BUFFER
                                        ;
001126  012767  000600  176422          MOV     #<Q$CHB+Q$CHRL>,@R$CR2  ;CLEAR OPERATIONAL INTERRUPTS
```

```
 287 001134  012767  001000  176422·          MOV·    #Q$REBK,QR$CR2          ;RE-ARM· INTERRUPTS·
 288                                ;
 289 001142· 012700  000000G·                 MOV·    #HLB,R0                 ;POINT·TO·HIT·BUFFER
 290 001146  016701  000000G·                 MOV·    BCL,R1                  ;LOAD·NUMBER·OF·WORDS·IN·HIT·BUFFER·
 291 001152· 005020                  1$:      CLR·    (R0)+                   ;CLEAR·HIT LIST·
 292·001154  005301                           DEC·    R1
 293 001156  001375                           BNE·    1$
 294                                ;
 295                                ;       IF A·BREAKPOINT·IS·IN·EFFECT, BRANCH·TO·CODE·THAT·PREPARES·
 296                                ;       TO FIELD·A·BREAKPOINT·INTERRUPT. ELSE·START·FREE·RUN WITHOUT·
 297                                ;       BREAKPOINTS.
 298                                ;
 299 001160  032767  000000G·000000G·         BIT·    #BREAK,BASE·            ;ANY·BREAKPOINTS·SET
 300 001166  001073                           BNE·    CBK2                    ;YES, SET·UP·FOR·BREAKPOINT·
 301 001170  012767  120000  176422·          MOV·    #<Q$SM+Q$ENOP>,QR$CR2·  ;ENABLE·OPERATIONAL·INTERRUPTS
 302·001176  012746  000360                   MOV·    #<Q$CSEL>,-(SP)         ;CLR·SELECT·BITS·
 303 001202· 052716  001001·                  BIS·    #<Q$LBD+Q$LBP>,(SP)     ;CLEAR·DRIVE·AND·PULSE·
 304 001206  052716  176000                   BIS·    #<Q$NCLK>,(SP)          ;CLEAR·NO-CLOCKS·
 305 001212· 005046                           CLR·    -(SP)                   ;SET·NOTHING·
 306 001214                                   CALL·   CSR1                    ;RE-WRITE·CSR1
 307                                ;
 308                                ;       SEND·ATTENTION·CODE·AND·BCL·ADDRESS·
 309                                ;
 310 001220  012767  000001  176424           MOV·    #Q$BCL,QR$LBR·          ;SET·ATTN·CODE·=·1
 311 001226  012767  120100  176422·          MOV·    #<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2·;SET·ATTN·CODE·READY
 312·001234  016701  176422·          11$:    MOV·    QR$CR2,R1               ;READ CSR2
 313 001240  032701  000100·                  BIT·    #Q$ATTN,R1              ;IS·ATTN·BIT·CLEAR·
 314 001244  001373                           BNE·    11$                     ;NO, LOOP·UNTIL·CLEAR·
 315                                ;
 316 001246  012767  000000G·176424           MOV·    #BCL,QR$LBR·            ;MOVE·BCL·ADDR·TO·LOD·BUS·REG·
 317 001254  012767  120040  176422·          MOV·    #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2·;SET·CC·TO·CP·SLAVE·DATA·READY·
 318 001262· 016701  176422           12$:    MOV·    QR$CR2,R1               ;READ·CSR2
 319 001266  032701  000040·                  BIT·    #Q$CCCP,R1              ;IS·CC·TO·CP·CLEAR·
 320 001272· 001373                           BNE·    12$                     ;NO, LOOP·UNTIL·CLEAR·
 321                                ;
 322·                               ;       PROVIDE·FOR·INTERRUPT·FROM·TERMINAL·TO·RETURN·CONTROL·
 323                                ;       TO USER·(IN·CASE·MICROCODE·HANGS UP·OR·RUNS·FOREVER.)
 324                                ;       CALL·SUBROUTINE·THAT·WAITS·FOR·INTERRUPTS·THEM·DECODES·
 325                                ;       THEM.
 326                                ;
 327 001274                                   CLEF$S· #EFN.4                  ;CLEAR·QIO EVENT·FLAG·
 328                                ;
 329 001306                                   CALL·   HANG2·                  ;ISSUE·QIO TO·TERMINAL·
 330 001312·                                  CALL·   WINT                    ;WAIT·FOR·EVENTS·
 331 001316                                   CALL·   KILL                    ;KILL·QIO·
 332·                               ;
 333                                ;       TURN·OFF·CLOCKS, SET·LOAD·MODE, PRINT·MICROPGM·ADDRESS·
 334                                ;
 335 001322· 005046                           CLR·    -(SP)                   ;CLEAR·NOTHING·
 336 001324  012746  176000·                  MOV·    #<Q$NCLK>,-(SP)         ;SET·NO-CLOCKS·
 337 001330                                   CALL·   CSR1                    ;WRITE·CSR 1
 338 001334  005046                           CLR·    -(SP)                   ;SET·CP·CONTROL·REG·TO·ZERO·
 339 001336                                   CALL·   CPCR                    ;DO·IT·
 340 001342· 005067  176422                   CLR·    QR$CR2·                 ;SET·LOAD·MODE·
 341 001346                                   CALL·   CPMP                    ;READ·CP·MICROPGM·ADDRESS·
 342·001352· 000167  000000G·                 JMP·    CPXX
 343                                ;
```

```
                                ;
                                ;       EXPECT TO HIT A BREAKPOINT.
                                ;
001356                          CBK2:
001356                                  CLEF$S  #EFN.3                  ;CLEAR BREAKPOINT EVENT FLAG
                                ;
001370                                  CLEF$S  #EFN.4                  ;CLEAR QIO EVENT FLAG
                                ;
                                ;       SET BKPT ALLOW BIT IN CP CONTROL REG
                                ;       SET SEARCH MODE
                                ;       SET FREE RUN CLOCK
                                ;
001402  012746  001000                  MOV     #CBKALLOW,-(SP)         ;SET CP CONTROL BIT
001406                                  CALL    CPCR                    ;WRITE CP CONTROL WORD
001412  012767  160000  176422          MOV     #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2    ;SET SEARCH MODE
001420  012746  000360                  MOV     #<Q$CSEL>,-(SP)         ;CLR SELECT BITS        FW
001424  052716  001001                  BIS     #<Q$LBD+Q$LBP>,(SP)     ;CLEAR DRIVE AND PULSE  FW
001430  052716  176000                  BIS     #<Q$NCLK>,(SP)          ;CLEAR NO-CLOCKS
001434  005046                          CLR     -(SP)                   ;SET NOTHING                    FW
001436                                  CALL    CSR1                    ;RE-WRITE CSR1                  FW
                                ;
                                ;       SEND BCL ADDRESS TO CP.
                                ;
001442  012767  000001  176424          MOV     #Q$BCL,QR$LBR           ;SET ATTN CODE = 1
001450  012767  160100  176422          MOV     #<Q$ATTN+Q$SM+Q$ENBK+Q$ENOP>,QR$CR2 ;SET ATTN CODE READY
001456  016701  176422          11$:    MOV     QR$CR2,R1               ;READ CSR2
001462  032701  000100                  BIT     #Q$ATTN,R1              ;IS ATTN BIT CLEAR
001466  001373                          BNE     11$                     ;NO, LOOP UNTIL CLEAR
                                ;
001470  012767  000000G 176424          MOV     #BCL,QR$LBR             ;MOVE BCL ADDR TO LOD BUS REG
001476  012767  160040  176422          MOV     #<Q$CCCP+Q$SM+Q$ENBK+Q$ENOP>,QR$CR2 ;SET CC TO CP SLAVE DATA READY
001504  016701  176422          12$:    MOV     QR$CR2,R1               ;READ CSR2
001510  032701  000040                  BIT     #Q$CCCP,R1              ;IS CC TO CP CLEAR
001514  001373                          BNE     12$                     ;NO, LOOP UNTIL CLEAR
                                ;
                                ;       PROVIDE FOR INTERRUPT FROM TERMINAL TO RETURN CONTROL
                                ;       TO USER (IN CASE MICROCODE HANGS UP OR RUNS FOREVER.)
                                ;       CALL SUBROUTINE THAT WAITS FOR INTERRUPTS THEM DECODES
                                ;       THEM.
                                ;
001516                                  CALL    HANG2                   ;KILL QIO
001522                                  CALL    WINT                    ;WAIT FOR INTERRUPTS
                                ;
                                ;       RESET NO-CLOCKS IN CSR 1
                                ;       SET LOAD MODE
                                ;       PRINT MICROPGM ADDRESS
                                ;
001526  005046                          CLR     -(SP)                   ;CLEAR NOTHING
001530  012746  176000                  MOV     #<Q$NCLK>,-(SP)         ;SET NO-CLOCKS
001534                                  CALL    CSR1                    ;RE-WRITE CSR1
001540  005046                          CLR     -(SP)                   ;SEND 0 TO CP CONTROL REG
001542                                  CALL    CPCR                    ;DO IT
                                ;
001546  005067  176422                  CLR     QR$CR2                  ;SET LOAD MODE
001552  012767  001000  176422          MOV     #Q$REBK,QR$CR2          ;RE-ARM INTERRUPTS
001560                                  CALL    CPMP                    ;PRINT MICROPGM ADDRESS
001564                                  CALL    KILL                    ;KILL AST
```

```
    401 001570  000167  000000G·····              JMP···· CPXX
```

```
                                ;
                                ;
                                ;       WAIT FOR INTERRUPTS.
                                ;       NB. DO NOT USE 'WTSE$S' OR 'WTLO$S'; THEY CAUSE SYSTEM PROBLEMS.
                                ;       IN CONNECTION WITH THE WAY INTERRUPTS ARE HANDLED (SEE THE
                                ;       'BPTISR' SUBROUTINE IN QMAIN).
                                ;
                                ;
001574                          WINT:
001574                          100$:   RDAF$S  #EFBUF                  ;READ EVENT FLAGS.
                                ;
001606  032767  000014  000000G         BIT     #<BIT3+BIT2>,EFBUF      ;ANY INTERRUPTS.
001614  001767                          BEQ     100$                    ;NO, READ AGAIN.
                                ;
                                ;       INTERRUPT DETECTED, DECODE IT.
                                ;       NB. EVENT FLAGS NUMBERS BEGIN WITH 1, BIT NUMBERS IN THE
                                ;       EVENT FLAG BUFFER BEGIN WITH 0.
                                ;       IF THE INTERRUPT IS FROM THE TERMINAL, IMMEDIATELY
                                ;       EXIT THIS SUBROUTINE.
                                ;
001616  032767  000010  000000G         BIT     #BIT3,EFBUF             ;EF #4 = INTERRUPT FROM TERMINAL.
001624  001402                          BEQ     10$                     ;NO.
001626  000167  000420                  JMP     WINTX2                  ;EXIT IF TRUE.
                                ;
                                ;       READ CSR2. CSR2 INCLUDES THE INTERRUPT FLAGS.
                                ;
001632  016767  176422  000000G 10$:     MOV     @R$CR2,APLACE           ;READ CSR #2.
001640  042767  177770  000000G         BIC     #177770,APLACE          ;MASK OFF INTERRUPT BITS.
                                ;
                                ;
                                ;       CHECK FOR CP BREAKPOINT INTERRUPT.
                                ;       BECAUSE BREAKPOINT INTERRUPTS ARE OF THE HIGHEST PRIORITY,
                                ;       CHECK TO SEE WHETHER A HIT LIST OR HRL INTERRUPT IS PENDING.
                                ;       THIS WILL ONLY HAPPEN WHEN IN THE MICROCODE A BREAKPOINT
                                ;       HALT IMMEDIATELY FOLLOWS AN HL OR HRL INTERRUPT ('CPBUG2'
                                ;       WILL SEE THEM IN REVERSE ORDER).
                                ;
                                ;       DMA INTERRUPTS WILL GO UNREPORTED UNDER THESE CIRCUMSTANCES.
                                ;
                                ;
                                ;       TELL WHETHER A HIT BUFFER INTERRUPT IS PENDING BY LOOKING
                                ;       AT THE BCL FOR A CLEARED ENTRY.
                                ;
001646  122767  000006  000000G         CMPB    #Q$ICP,APLACE           ;CP BREAKPOINT INTERRUPT.
001654  001060                          BNE     OTHERS                  ;NO, CHECK OTHERS.
001656  005767  000002G                 TST     BCL+2                   ;HIT BUFFER BCL ENTRY CLEARED.
001662  001025                          BNE     2$                      ;NO, TEST HRL.
001664  012767  101000  176422          MOV     #<Q$SM+Q$REBK>,@R$CR2   ;RE-ARM INTERRUPTS.
001672  012767  120000  176422          MOV     #<Q$SM+Q$ENOP>,@R$CR2   ;ENABLE OPERATIONAL INTERRUPTS.
001700  012767  100600  176422          MOV     #<Q$SM+Q$CHB+Q$CHRL>,@R$CR2 ;CLEAR OPERATIONAL INTERRUPTS.
001706  012767  000000G 000002G         MOV     #HLB,BCL+2              ;RESTORE BCL ENTRY.
001714  012700  000017'                 MOV     #HBMSG,R0               ;POINT TO HIT LIST MESSAGE.
001720  012705  000000G                 MOV     #PRINT,R5               ;POINT TO PRINT LINE
001724  112025                  1$:     MOVB    (R0)+,(R5)+
001726  001376                          BNE     1$
001730                                  CALL    CONSOL                  ;PRINT MESSAGE.
001734  000541                          BR      WINTX                   ;LEAVE INTERRUPT ROUTINE.
```

```
                                        ;
                                        ;       CHECK WHETHER AN HRL ENTRY IS PENDING BY LOOKING
                                        ;       FOR A CLEARED BCL ENTRY.
                                        ;
001736  005767  000012G         2$:     TST     BCL+10.                 ;HRL BCL ENTRY CLEARED
001742  001136                  .       BNE     WINTX                   ;NO, EXIT
001744  012767  101000  176422          MOV     #<Q$SM+Q$REBK>,QR$CR2   ;RE-ARM INTERRUPTS
001752  012767  120000  176422          MOV     #<Q$SM+Q$ENOP>,QR$CR2   ;ENABLE OPERATIONAL INTERRUPTS
001760  012767  100600  176422          MOV     #<Q$SM+Q$CHB+Q$CHRL>,QR$CR2 ;CLEAR OPERATIONAL INTERRUPTS
001766  012767  000000G 000012G         MOV     #HRL,BCL+10.            ;RESTORE BCL ENTRY
001774  012700  000042'                 MOV     #HRLMSG,R0              ;POINT TO HIT LIST MESSAGE
002000  012705  000000G                 MOV     #PRINT,R5               ;POINT TO PRINT LINE
002004  112025                  3$:     MOVB    (R0)+,(R5)+
002006  001376                          BNE     3$
002010                                  CALL    CONSOL                  ;PRINT MESSAGE
002014  000511                          BR      WINTX                   ;LEAVE INTERRUPT ROUTINE
                                        ;
                                        ;
                                        ;       CHECK FOR MRP INTERRUPT
                                        ;
002016  122767  000007  000000G OTHERS: CMPB    #Q$IMRP,APLACE          ;MRP BREAKPOINT ?
002024  001003                          BNE     1$                      ;NO, TRY NEXT
002026  012700  000000'                 MOV     #MRPMSG,R0              ;R0 -> MESSAGE
002032  000443                          BR      PMSG                    ;PRINT MESSAGE
                                        ;
                                        ;       CHECK FOR HIT BUFFER INTERRUPT
                                        ;
002034  122767  000003  000000G 1$:     CMPB    #Q$IHB,APLACE           ;HIT LIST INTERRUPT
002042  001011                          BNE     2$                      ;NO, TRY NEXT
002044  012767  100400  176422          MOV     #<Q$SM+Q$CHB>,QR$CR2    ;CLEAR HL INTERRUPT
002052  012767  000000G 000002G         MOV     #HLB,BCL+2              ;PUT BUFFER ADDRESS BACK IN BCL
002060  012700  000017'                 MOV     #HBMSG,R0               ;POINT TO HIT LIST MESSAGE
002064  000426                          BR      PMSG                    ;PRINT MESSAGE
                                        ;
                                        ;       CHECK FOR HRL INTERRUPT
                                        ;
002066  122767  000002  000000G 2$:     CMPB    #Q$IHRL,APLACE          ;HRL INTERRUPT
002074  001011                          BNE     3$                      ;NO, TRY NEXT
002076  012767  100200  176422          MOV     #<Q$SM+Q$CHRL>,QR$CR2   ;CLEAR HRL INTERRUPT
002104  012767  000000G 000012G         MOV     #HRL,BCL+10.            ;PUT BUFFER ADDRESS BACK IN BCL
002112  012700  000042'                 MOV     #HRLMSG,R0              ;POINT TO HRL MESSAGE
002116  000411                          BR      PMSG
                                        ;
                                        ;       CHECK FOR DMA INTERRUPT
                                        ;       NB. DMA USES HIT BUFFER INTERRUPT REQUEST FROM CP
                                        ;       (SEE CP MICROCODE).
                                        ;
002120  122767  000001  000000G 3$:     CMPB    #Q$DMA,APLACE           ;DMA ?
002126  001030                          BNE     UNID
002130  012767  100400  176422          MOV     #<Q$SM+Q$CHB>,QR$CR2    ;CLEAR DMA INTERRUPT (USING HIT BUF FLAG)
002136  012700  000060'                 MOV     #DMAMSG,R0              ;POINT TO DMA MESSAGE
                                        ;
                                        ;       PRINT MESSAGE, CLEAR INTERRUPT FLAG, WAIT FOR NEXT INTERRUPT.
                                        ;
002142  012705  000000G         PMSG:   MOV     #PRINT,R5               ;POINT TO PRINT LINE
002146  112025                  3$:     MOVB    (R0)+,(R5)+
002150  001376                          BNE     3$
```

```
                                        CALL·   CONSOL·                 ;WRITE·MESSAGE·TO·TERMINAL·
                                ;
                                        CLEF$S· #EFN.3                  ;CLEAR·HOR INTERRUPT.EVENT.FLAG·
                                ;
                                        MOV·    #<Q$SM+Q$REBK>,QR$CR2·  ;RE-ARM·INTERRUPTS·
                                        MOV·    #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2·;ENABLE·INTERRUPTS·
                                        JMP·    WINT                    ;WAIT·FOR·NEXT·INTERRUPT·
                                ;
                                ;       UNIDENTIFIED INTERRUPT
                                ;
                                UNID:   MOV·    #UNMSG,R0               ;POINT·TO·MESSAGE·
                                        MOV·    #PRINT,R5               ;POINT·TO·PRINT·LINE
                                5$:     MOVB    (R0)+,(R5)+
                                        BNE·    5$
                                        MOV·    APLACE,R1               ;LOAD·CSR2 INTO·R1
                                        CALL·   UNPK                    ;CONVERT·TO·ASCII·HEX·
                                        CALL·   CONSOL·                 ;WRITE·MESSAGE·TO·TERMINAL·
                                ;
                                ;       EXIT·
                                ;
                                WINTX:
                                        CLEF$S· #EFN.3                  ;CLEAR·HOR INTERRUPT EVENT·FLAG·
                                ;
                                WINTX2: RETURN·
```

```
                                ;
                                ;
                                ;       PRINT·CP·MICROPROGRAM·ADDRESS·
                                ;
                                ;
002254                          CPMP:
002254  012746  000004                  MOV·    #<CSADRD>,-(SP)         ;READ MPA·SELECT·BIT
002260                                  CALL·   CPCR                    ;WRITE·CP·CONTROL·REGISTER·
002264                                  CALL·   CPLB                    ;REQUEST·CP·TO·LOD·BUS·
002270  012601                          MOV·    (SP)+,R1                ;UNLOAD·MPA·
002272· 042701  176000                  BIC·    #176000,R1              ;CLEAR·UPPER·6 BITS·
002276  012705  000000G·                MOV·    #PRINT,R5               ;POINT·TO·PRINT·LINE
002302·                                 CALL·   UNPK                    ;CONVERT·MPA·*
002306                                  CALL·   CONSOL·                 ;WRITE·TO·CONSOLE·
002312· 005046                          CLR·    -(SP)                   ;CLEAR·CP·CR·
002314                                  CALL·   CPCR
002320                                  RETURN·
```

```
                                        ;
                                        ;
                                        ;       INITIALIZE·PPS·
                                        ;
                                        ;
                                        ;       SET·HP·QUIESCENT·
                                        ;
002322·                                 PPINIT:
002322· 012746  000053                          MOV·    #Q$QLA,-(SP)            ;ADDRESS·SELECT·FOR·QLB·PAGES·
002326                                          CALL·   PPCR
002332· 012746  002000                          MOV·    #2000,-(SP)             ;SEND·ADDRESS·X'400'  (ILLEGAL)
002336                                          CALL·   LBPP
                                        ;
                                        ;       RESET·MRP·AND·CP·
                                        ;
002342· 005046                                  CLR·    -(SP)                   ;CLEAR·NOTHING·IN·CSR1
002344  012746  000004                          MOV·    #Q$MSET,-(SP)           ;SET·RESET
002350                                          CALL·   CSR1
002354  012746  000004·                         MOV·    #Q$MSET,-(SP)           ;CLEAR·RESET·
002360  005046                                  CLR·    -(SP)                   ;SET·NOTHING·
002362·                                         CALL·   CSR1
                                        ;
002366  005046                                  CLR·    -(SP)                   ;CLEAR·NOTHING·IN·CSR1
002370  012746  000002                          MOV·    #Q$CSET,-(SP)           ;SET·RESET
002374                                          CALL·   CSR1
002400  012746  000002·                         MOV·    #Q$CSET,-(SP)           ;CLEAR·RESET·
002404  005046                                  CLR·    -(SP)                   ;SET·NOTHING·
002406                                          CALL·   CSR1
                                        ;
                                        ;       SET·'QLB·ERASE'
                                        ;
002412· 032767  000000G·000000G·                BIT·    #NEQLB,BASE·            ;NO·ERASE·FOR·QLB·?·
002420· 001406                                  BEQ·    1$                      ;ERASE·QLB
002422· 042767  000000G·000000G·                BIC·    #NEQLB,BASE·            ;CLEAR·NO·ERASE·FLAG
002430  012746  000300                          MOV·    #300,-(SP)              ;SET·CODE·=·X'C0'
002434  000402·                                 BR      2$
                                        ;
002436  012746  000100                  1$:     MOV·    #100,-(SP)              ;SEND·X'40'  TO·PPCR·
002442·                                 2$:     CALL·   PPCR
002446                                          RETURN·
                                        ;
000001                                          .END·
```

```
ALUCKE=·040000          BYTE38=·000046          BYTE9 =·000011          LBPP· =·****** GX·      Q$FC  =·000045
ALUOE·=·004000          BYTE39=·000047          BYTE90=·000132·         LBPSC·=·****** GX·      Q$FO  =·000044
APLACE=·****** GX·      BYTE4 =·000004          BYTE91=·000133          LOC.EN=·000100          Q$FP  =·000046
A01   =·010000          BYTE40=·000050          BYTE92=·000134          LOC.WA=·040000          Q$HBF =·000002
BASE· =·****** GX·      BYTE41=·000051          BYTE93=·000135          LOC.WB=·100000          Q$ICP =·000006
BCL·  =·****** GX·      BYTE42=·000052·         BYTE94=·000136          MAREN1=·000001          Q$IHB =·000003
BINWD·=·****** GX·      BYTE43=·000053          BYTE95=·000137          MAREN2=·004000          Q$IHRL=·000002
BITVAL=·000000          BYTE44=·000054          BYTE96=·000140          MARLOD=·010000          Q$IMRP=·000007
BIT0  =·000001          BYTE45=·000055          BYTE97=·000141          MAROUT=·000002·         Q$LBD =·001000
BIT1  =·000002·         BYTE46=·000056          BYTE98=·000142·         MAR.LO=·002000          Q$LBDP=·001001
BIT10 =·002000          BYTE47=·000057          BYTE99=·000143          MAR.OU=·000040          Q$LBP =·000001
BIT11 =·004000          BYTE48=·000060          BYTVAL=·000144          MBKALL=·001000          Q$LDCD=·000003
BIT12·=·010000          BYTE49=·000061          CBK·     000546R·   002·MBKCLK=·000400          Q$LDMD=·000004
BIT13 =·020000          BYTE5 =·000005          CBKALL=·001000          MMADRD=·000100          Q$LDPP=·002000
BIT14 =·040000          BYTE50=·000062·         CBKCLK=·000400          MMLEFT=·000002·         Q$LHP =·010000
BIT15 =·100000          BYTE51=·000063          CBK2·    001356R·   002·MMOE· =·000004          Q$MNC =·140000
BIT2· =·000004          BYTE52=·000064          CGO·     000252R·   002·MMWRTE=·000010          Q$MR  =·000052
BIT3  =·000010          BYTE53=·000065          CNOBRE=·100000          MNOBRE=·100000          Q$MRP =·000040
BIT4  =·000020          BYTE54=·000066          CONSOL=·****** GX·      MREN1 =·000001          Q$MRP2=·000240
BIT5  =·000040          BYTE55=·000067          CPCCEN=·010000          MREN2 =·020000          Q$MSC =·040000
BIT6  =·000100          BYTE56=·000070          CPCR· =·****** GX·      MRPCR =·****** GX·      Q$MSET=·000004
BIT7  =·000200          BYTE57=·000071          CPLB· =·****** GX·      MRPMSG·  000000R·   002·Q$MSP =·100000
BIT8·=·000400           BYTE58=·000072·         CPMP·    002254R·   002·MSYN· =·000040          Q$NCLK=·176000
BIT9  =·001000          BYTE59=·000073          CPREAD=·040000          N·      =·000144        Q$PP  =·000100
BREAK·=·****** GX·      BYTE6 =·000006          CPWRTE=·020000          NEQLB·=·****** GX·      Q$PPSW=·000320
BYTE0 =·000000          BYTE60=·000074          CPXX· =·****** GX·      OTHERS·  002016R·   002·Q$PP2·=·000300
BYTE1 =·000001          BYTE61=·000075          CSADRD=·000004          PACK· =·****** GX·      Q$QHLT=·000013
BYTE10=·000012·         BYTE62=·000076          CSEQCI=·100000          PLB·  =·000010          Q$QL  =·000043
BYTE11=·000013          BYTE63=·000077          CSHIGH=·****** GX·      PLC·  =·000020          Q$QLA·=·000053
BYTE12=·000014          BYTE64=·000100          CSOE· =·000040          PLD·  =·000030          Q$QLB·=·000054
BYTE13=·000015          BYTE65=·000101          CSR1· =·****** GX·      PLRWR·=·000200          Q$QLR·=·000001
BYTE14=·000016          BYTE66=·000102·         CSWRTE=·000100          PLR.EN=·000200          Q$QW  =·000042
BYTE15=·000017          BYTE67=·000103          DBR.RD=·000001          PMSG·    002142R·   002·Q$RDCD=·000005
BYTE16=·000020          BYTE68=·000104          DB$CPP=·001457          PPCR· =·****** GX·      Q$RDMD=·000006
BYTE17=·000021          BYTE69=·000105          DB$SPT=·000026          PPINIT·  002322R·   002·Q$REBK=·001000
BYTE18=·000022·         BYTE7 =·000007          DB$TPC=·000023          PRINT·=·****** GX·      Q$RNC·=·006000
BYTE19=·000023          BYTE70=·000106          DISPGS=·100000          QR$CR1=·176420          Q$RSC·=·004000
BYTE2·=·000002·         BYTE71=·000107          DMAAWR=·000005          QR$CR2=·176422·         Q$RSET=·000010
BYTE20=·000024          BYTE72=·000110          DMAMSG·  000060R·   002·QR$LBR=·176424          Q$SM  =·100000
BYTE21=·000025          BYTE73=·000111          DMARRD=·000003          Q$ATTN=·000100          Q$SP  =·000120
BYTE22=·000026          BYTE74=·000112·         DMARWR=·000004          Q$BCL·=·000001          Q$SP2·=·000340
BYTE23=·000027          BYTE75=·000113          EFBUF·=·****** GX·      Q$CCCP=·000040          RGO.EN=·000200
BYTE24=·000030          BYTE76=·000114          EFN.3 =·****** GX·      Q$CHB·=·000400          RGO.VA=·020000
BYTE25=·000031          BYTE77=·000115          EFN.4 =·****** GX·      Q$CHRL=·000200          SEQCS·=·****** GX·
BYTE26=·000032·         BYTE78=·000116          ENBR·  =·010000         Q$CLR·=·000040          SEQMM·=·****** GX·
BYTE27=·000033          BYTE79=·000117          ERR10 =·****** GX·      Q$CNC·=·030000          SEQ.CI=·000010
BYTE28=·000034          BYTE8 =·000010          ERR12·=·****** GX·      Q$CP· =·000060          S$CLR·=·000000
BYTE29=·000035          BYTE80=·000120          ERR5  =·****** GX·      Q$CPCC=·000010          S$LA  =·000001
BYTE3 =·000003          BYTE81=·000121          FIND·  =·****** GX·     Q$CP2·=·000260          S$OB  =·000005
BYTE30=·000036          BYTE82=·000122·         FR2·     000770RG·  002·Q$CSC·=·010000          S$OR  =·000006
BYTE31=·000037          BYTE83=·000123          GO2·     000136RG·  002·Q$CSEL=·000360          S$OX  =·000004
BYTE32=·000040          BYTE84=·000124          HANG2·=·****** GX·      Q$CSET=·000002·         S$SR  =·000007
BYTE33=·000041          BYTE85=·000125          HBMSG·   000017R·   002·Q$CSP·=·020000          S$SI  =·000010
BYTE34=·000042·         BYTE86=·000126          HLB·   =·****** GX·     Q$DMA·=·000004          S$S2  =·000014
BYTE35=·000043          BYTE87=·000127          HRL·   =·****** GX·     Q$ENBK=·040000          TD$CTR=·176370
BYTE36=·000044          BYTE88=·000130          HRLMSG·  000042R·   002·Q$ENOP=·020000          TD$CTW=·176360
BYTE37=·000045          BYTE89=·000131          KILL·  =·****** GX·     Q$FAL·=·004000          TD$INL=·004000
```

```
CPBUG2  M[illegible]O M1110  27-MAR-80 14:42  PAGE 10-2
SYMBOL TA[illegible]
```

```
TD$MEM= 000270          T$OBE  = 004000          WORD16= 000040          WORD45= 000132          WORD74= 000224
TD$OAR= 176344          T$OBF  = 010000          WORD17= 000042          WORD46= 000134          WORD75= 000226
TD$OTR= 176346          T$OBRA= 000034           WORD18= 000044          WORD47= 000136          WORD76= 000230
TD$ORD= 000274          T$OBWA= 000032           WORD19= 000046          WORD48= 000140          WORD77= 000232
TD$SW  = 176376         T$OUTA= 100000           WORD2  = 000004         WORD49= 000142          WORD78= 000234
TD$TAR= 176372          T$RBDO= 000200           WORD20= 000050          WORD5  = 000012         WORD79= 000236
TD$TAW= 176362          T$RNB  = 000040          WORD21= 000052          WORD50= 000144          WORD8  = 000020
TD$TDR= 176374          T$RSET= 040000           WORD22= 000054          WORD51= 000146          WORD80= 000240
TD$TDW= 176364          T$SC   = 000022          WORD23= 000056          WORD52= 000150          WORD81= 000242
T$AD   = 000020         T$SCLK= 020000           WORD24= 000060          WORD53= 000152          WORD82= 000244
T$BA   = 000002         T$SEG1= 000000           WORD25= 000062          WORD54= 000154          WORD83= 000246
T$BD   = 000010         T$SEG2= 000001           WORD26= 000064          WORD55= 000156          WORD84= 000250
T$BSO  = 100000         T$SEG3= 000002           WORD27= 000066          WORD56= 000160          WORD85= 000252
T$BT   = 000020         T$SO   = 000001          WORD28= 000070          WORD57= 000162          WORD86= 000254
T$BTAR= 000030          T$UBUS= 100000           WORD29= 000072          WORD58= 000164          WORD87= 000256
T$BTD  = 002000         T$1CLK= 000400           WORD3  = 000006         WORD59= 000166          WORD88= 000260
T$CD   = 000100         T$8BEN= 000020           WORD30= 000074          WORD6  = 000014         WORD89= 000262
T$CLK  = 002000         UBD.IN= 000020           WORD31= 000076          WORD60= 000170          WORD9  = 000022
T$DISK= 000200          UNID   002210R     002  WORD32= 000100          WORD61= 000172          WORD90= 000264
T$DRD  = 000004         UNMSG  000076R     002  WORD33= 000102          WORD62= 000174          WORD91= 000266
T$EMEM= 010000          UNPK   = ****** GX       WORD34= 000104          WORD63= 000176          WORD92= 000270
T$FSAA= 000000          WINT   001574R     002  WORD35= 000106          WORD64= 000200          WORD93= 000272
T$FSAB= 000004          WINTX  002240R     002  WORD36= 000110          WORD65= 000202          WORD94= 000274
T$FSAC= 000014          WINTX2 002252R     002  WORD37= 000112          WORD66= 000204          WORD95= 000276
T$FSB2= 000010          WORD0  = 000000          WORD38= 000114          WORD67= 000206          WORD96= 000300
T$IB   = 000026         WORD1  = 000002          WORD39= 000116          WORD68= 000210          WORD97= 000302
T$IBAR= 000024          WORD10= 000024           WORD4  = 000010         WORD69= 000212          WORD98= 000304
T$IBE  = 020000         WORD11= 000026           WORD40= 000120          WORD7  = 000016         WORD99= 000306
T$IBF  = 040000         WORD12= 000030           WORD41= 000122          WORD70= 000214          WRDVAL= 000310
T$ICD  = 000040         WORD13= 000032           WORD42= 000124          WORD71= 000216          XTREAD= 001000
T$MODE= 004000          WORD14= 000034           WORD43= 000126          WORD72= 000220          XTWRTE= 000400
T$OB   = 000036         WORD15= 000036           WORD44= 000130          WORD73= 000222
```

```
. ABS.  000000     000
        000000     001
CPBUG2  002450     002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3902 WORDS  ( 16 PAGES)
DYNAMIC MEMORY:  4916 WORDS  ( 18 PAGES)
ELAPSED TIME:  00:00:58
CPBUG2,CPBUG2/-SP=[20,1]IM,[20,1]CPBUG2
```

```
                                        .TITLE CPREST
                                        .PSECT CPREST
                                ;
                                ;
                                ;       HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
                                ;       CONTROL PROCESSOR TEST ROUTINES
                                ;
                                ;       COMMANDS:
                                ;       ST      STORE INTO AN CP REGISTER
                                ;       RE      READ FROM A REGISTER
                                ;       RS      RESET CP
                                ;       CL      CALL HQR LOADER
                                ;
                                ;       ONCE A COMMAND HAS BEEN EXECUTED (OR AN ERROR ENCOUNTERED)
                                ;       THIS MODULE RETURNS CONTROL TO THE MODULE CP AT LOCATION
                                ;       'CPXX'.
                                ;
                                ;
                                ;       STORE
                                ;       TABLE OF VALID REGISTER MNEMONICS AND ASSOCIATED ROUTINE
                                ;       ADDRESSES
                                ;
000000                          ST2TBL::
000000      103     122                 .ASCII  /CR/                    ;CP CONTROL REG
000002  000400'                         .WORD   ST2CR
000004      102     113                 .ASCII  /BK/                    ;BREAKPOINT REG
000006  000414'                         .WORD   ST2BK
000010      115     101                 .ASCII  /MA/                    ;MEMORY ADDRESS REG
000012  000440'                         .WORD   ST2MA
000014      120     101                 .ASCII  /PA/                    ;PIPELINE REG A SECT
000016  000530'                         .WORD   ST2PA
000020      120     102                 .ASCII  /PB/                    ;PIPELINE REG B SECT
000022  000562'                         .WORD   ST2PB
000024      120     103                 .ASCII  /PC/                    ;PIPELINE REG C SECT
000026  000614'                         .WORD   ST2PC
000030      120     104                 .ASCII  /PD/                    ;PIPELINE REG D SECT
000032  000646'                         .WORD   ST2PD
000034      104     101                 .ASCII  /DA/                    ;DMA ADDRESS REGISTER
000036  000700'                         .WORD   ST2DA
000040      104     104                 .ASCII  /DD/                    ;DMA DATA REGISTER
000042  000742'                         .WORD   ST2DD
000044      122.    060                 .ASCII  /R0/                    ;CP REGISTER 0
000046  001100'                         .WORD   ST2R0
000050      122.    061                 .ASCII  /R1/
000052  001074'                         .WORD   ST2R1
000054      122.    062                 .ASCII  /R2/
000056  001070'                         .WORD   ST2R2
000060      122.    063                 .ASCII  /R3/
000062  001064'                         .WORD   ST2R3
000064      122.    064                 .ASCII  /R4/
000066  001060'                         .WORD   ST2R4
000070      122.    065                 .ASCII  /R5/
000072  001054'                         .WORD   ST2R5
000074      122.    066                 .ASCII  /R6/
000076  001050'                         .WORD   ST2R6
000100      122.    067                 .ASCII  /R7/
000102  001044'                         .WORD   ST2R7
```

```
 58 000104     122·     070                .ASCII·  /R8/
 59 000106  001040'                        .WORD·   ST2R8
 60 000110     122·     071                .ASCII·  /R9/
 61 000112· 001034'                        .WORD·   ST2R9
 62·000114     122·     101                .ASCII·  /RA/
 63 000116  001030'                        .WORD·   ST2RA·
 64 000120     122·     102                .ASCII·  /RB/
 65 000122· 001024'                        .WORD·   ST2RB·
 66 000124     122·     103                .ASCII·  /RC/
 67 000126  001020'                        .WORD·   ST2RC·
 68 000130     122·     104                .ASCII·  /RD/
 69 000132· 001014'                        .WORD·   ST2RD·
 70 000134     122·     105                .ASCII·  /RE/
 71 000136  001010'                        .WORD·   ST2RE·
 72·000140     122·     106                .ASCII·  /RF/
 73 000142· 001004'                        .WORD·   ST2RF·
 74          000031              ST2LN·    ==       <.-ST2TBL>/4
 75                              ;
 76                              ;         READ·
 77                              ;         TABLE·OF·VALID·MNEMONICS·AND·ASSOCIATED·ROUTINE·ADDRESSES·
 78                              ;
 79 000144                      RE2TBL::
 80 000144     115      101                .ASCII·  /MA/              ;READ·MEMORY·ADDRESS REG·
 81 000146  001230'                        .WORD·   RE2MA·
 82·000150     115      120                .ASCII·  /MP/              ;READ·MICROPGM·ADDRESS·
 83 000152· 001260'                        .WORD·   RE2MP·
 84 000154     104      104                .ASCII·  /DD/              ;READ·DMA·DATA·REG·
 85 000156  001314'                        .WORD·   RE2DD·
 86 000160     122·     060                .ASCII·  /R0/              ;CP·REGISTER·0
 87 000162· 001442'                        .WORD·   RE2R0
 88 000164     122·     061                .ASCII·  /R1/
 89 000166  001436'                        .WORD·   RE2R1
 90 000170     122·     062                .ASCII·  /R2/
 91 000172· 001432'                        .WORD·   RE2R2·
 92·000174     122·     063                .ASCII·  /R3/
 93 000176  001426'                        .WORD·   RE2R3
 94 000200     122·     064                .ASCII·  /R4/
 95 000202· 001422'                        .WORD·   RE2R4
 96 000204     122·     065                .ASCII·  /R5/
 97 000206  001416'                        .WORD·   RE2R5
 98 000210     122·     066                .ASCII·  /R6/
 99 000212· 001412'                        .WORD·   RE2R6
100 000214     122·     067                .ASCII·  /R7/
101 000216  001406'                        .WORD·   RE2R7
102·000220     122·     070                .ASCII·  /R8/
103 000222· 001402'                        .WORD·   RE2R8
104 000224     122·     071                .ASCII·  /R9/
105 000226  001376'                        .WORD·   RE2R9
106 000230     122·     101                .ASCII·  /RA/
107 000232· 001372'                        .WORD·   RE2RA·
108 000234     122·     102                .ASCII·  /RB/
109 000236  001366'                        .WORD·   RE2RB·
110 000240     122·     103                .ASCII·  /RC/
111 000242· 001362'                        .WORD·   RE2RC·
112·000244     122·     104                .ASCII·  /RD/
113 000246  001356'                        .WORD·   RE2RD·
114 000250     122·     105                .ASCII·  /RE/
```

```
000252  001352'                          .WORD   RE2RE
000254     122       106                 .ASCII  /RF/
000256  001346'                          .WORD   RE2RF
        000023                   RE2LN   ==      <.-RE2TBL>/4
                                 ;
                                 ;
                                 ;       STORE
                                 ;       PERFORM THIRD LEVEL PARSING
                                 ;       EG. IN THE COMMAND:
                                 ;       CP>ST 0 CR
                                 ;       PARSE THE '0'
                                 ;
                                 ;
000260                           ST2::
000260                                   CALL    FIND                    ;FIND OPERAND IN COMMAND LINE
000264  103004                           BCC     1$                      ;OK, CONTINUE
000266                                   CALL    ERR4                    ;NOTHING THERE
000272  000167   000636                  JMP     ST2X                    ;RETURN TO TOP OF LOOP (PROMPT)
000276                           1$:     CALL    PACK                    ;CONVERT VALUE IN COMMAND LINE TO BINARY
000302  103004                           BCC     2$
000304                                   CALL    ERR5
000310  000167   000620                  JMP     ST2X
                                 ;
                                 ;       CONTINUE PARSING
                                 ;       FIND THE REGISTER MNEMONIC IN THE COMMAND LINE
                                 ;
000314                           2$:     CALL    FIND                    ;LOCATE A NON-BLANK
000320  103004                           BCC     3$
000322                                   CALL    ERR4
000326  000167   000602                  JMP     ST2X
                                 ;
                                 ;       MATCH THE MNEMONIC FROM THE COMMAND LINE AGAINST THE
                                 ;       TABLE OF VALID MNEMONICS.
                                 ;
000332  012700   000031          3$:     MOV     #ST2LN,R0               ;NUMBER OF TABLE ENTRIES
000336  012702   000000'                 MOV     #ST2TBL,R2              ;R2 -> TABLE
000342                                   CALL    SCAN                    ;MATCH AGAINST COMMAND LINE
000346  103004                           BCC     4$                      ;OK, CONTINUE
000350                                   CALL    ERR6
000354  000167   000554                  JMP     ST2X
                                 ;
                                 ;       SAVE THE POINTER TO THE ROUTINE ASSOCIATED WITH THE
                                 ;       REGISTER. R1 -> ROUTINE ADDRESS.
                                 ;       CALL ROUTINE TO SCAN COMMAND LINE FOR LOOP INDICATOR.
                                 ;       EG. CP>ST 0 CR L
                                 ;       LOOP FLAG WILL BE SET IF INDICATOR IS PRESENT.
                                 ;       JUMP TO ROUTINE TO LOAD REGISTER.
                                 ;
                                 ;
000360  010167   000000G         4$:     MOV     R1,RTNPT                ;SAVE POINTER TO RTN
000364                                   CALL    LOOPR                   ;LOOP?
000370  016701   000000G         ST2IN:  MOV     RTNPT,R1                ;POINT TO ROUTINE
000374  000171   000000                  JMP     @(R1)                   ;EXECUTE ROUTINE
                                 ;
                                 ;
                                 ;       CP CONTROL REG
```

```
 172·000400                             ST2CR::
 173 000400  016746  000000G·                   MOV·    BINWD,-(SP)             ;MOVE·DATA·TO·LOD·BUS·REG·
 174 000404                                     CALL·   CPCR                    ;DIRECT·CNTL·WORD·TO·CP·
 175 000410  000167  000504                     JMP·    ST2LP·                  ;TEST·LOOP·FLAG·
 176                                    ;
 177                                    ;       BREAKPOINT·REG·
 178                                    ;
 179 000414                             ST2BK::
 180 000414  012746  000400                     MOV·    #CBKCLKEN,-(SP)         ;SET·CP·CNTL·FOR·BREAKPOINT·
 181 000420                                     CALL·   CPCR                    ;DIRECT·CNTL·WORD·TO·CP·
 182·000424  016746  000000G·                   MOV·    BINWD,-(SP)             ;LOAD·DATA·WORD·INTO·LOAD·BUS·REG·
 183 000430                                     CALL·   LBCP                    ;SET·BCE·CNTL·WORD·
 184 000434  000167  000460                     JMP·    ST2LP·                  ;TEST·LOOP·FLAG·
 185                                    ;
 186                                    ;       MEMORY·ADDRESS·REG·
 187                                    ;
 188 000440                             ST2MA::
 189 000440  012746  000230                     MOV·    #<PLRWR+PLD>,-(SP)
 190 000444                                     CALL·   CPCR                    ;DIRECT·CNTL·WORD·TO·CP·
 191 000450  012746  010000                     MOV·    #<MARLOD>,-(SP)         ;SEND·MAR·LOD·BIT·                      NS·
 192·000454                                     CALL·   LBCP                    ;SEND·DATA·TO·CP·
 193 000460  012746  000001                     MOV·    #<MREN1>, -(SP)         ;SET·MAREN·BIT·AND·ALSO·                NS·
 194 000464                                     CALL·   CPCR                    ;CLEAR·PLR-D·ENABLE·BITS·               NS·
 195 000470  016746  000000G·                   MOV·    BINWD,-(SP)             ;MOVE·DATA·WORD·TO·LOD·BUS·REG
 196 000474                                     CALL·   LBCP                    ;SEND·DATA·TO·CP·(MAR)
 197 000500  012746  000230                     MOV·    #<PLRWR+PLD>,-(SP)
 198 000504                                     CALL·   CPCR                    ;DIRECT·CNTL·WORD·TO·CP·CR·             NS·
 199 000510  005046                             CLR·    -(SP)                   ;CLEAR·PLR-D·BITS·                      NS·
 200 000512·                                    CALL·   LBCP                    ;SEND·DATA·TO·CP·                       NS·
 201 000516  005046                             CLR·    -(SP)                   ;CLEAR·CP·CR·BITS·                      NS·
 202·000520                                     CALL·   CPCR                    ; DIRECT·CNTL·WORD·TO·CP·CR·            NS·
 203 000524  000167  000370                     JMP·    ST2LP·
 204                                    ;
 205                                    ;       GP PIPELINE·REG·SECTION·A·
 206                                    ;
 207                                    ;       NOTE:   IF·CSR·#2·IS·SET·UP·FOR·SEARCH·MODE·FOR·ANY·REASON, THESE·
 208                                    ;               4 PIPELINE·ROUTINES·WILL NOT·WORK·!!!!
 209 000530                             ST2PA::
 210 000530  012746  000200                     MOV·    #PLRWR,-(SP)
 211 000534                                     CALL·   CPCR                    ;DIRECT·CNTL·WORD·TO·CP·
 212·000540  016746  000000G·                   MOV·    BINWD,-(SP)             ;MOVE·DATA·WORD·TO·LOD·BUS·REG
 213 000544                                     CALL·   LBCP                    ;SEND·DATA·TO·CP·
 214 000550  005046                             CLR·    -(SP)
 215 000552·                                    CALL·   CPCR                    ; CLEAR·THE·CONTROL·REGISTER·           NS·
 216 000556  000167  000336·                    JMP·    ST2LP·                  ;TEST·FOR·REPEAT·
 217                                    ;
 218                                    ;       GP PIPELINE·REG·SECTION·B·
 219                                    ;       CAUTION: DO·NOT·SET·BIT·14 DURING·THIS·TEST, IT·WILL LOCK·
 220                                    ;       UP THE·BUS·
 221                                    ;       ALSO·DO·NOT·USE·XXX1, XXX2, XXX6, XXX7, OR XXXA·
 222·                                   ;
 223 000562·                            ST2PB::
 224 000562·  012746  000210                    MOV·    #<PLRWR+PLB>,-(SP)
 225 000566                                     CALL·   CPCR                    ;DIRECT·CNTL·WORD·TO·CP·
 226 000572·  016746  000000G·                  MOV·    BINWD,-(SP)             ;MOVE·DATA·WORD·TO·LOD·BUS·REG
 227 000576                                     CALL·   LBCP                    ;SEND·DATA·TO·CP·
 228 000602·  005046                            CLR·    -(SP)                   ;                                       NS·
```

```
000604
000610  000167  000304                CALL    CPCR                    ; CLEAR THE CONTROL REGISTER    NS
                                      JMP     ST2LP                   ;TEST FOR REPEAT
                                ;
                                ;       CP PIPELINE REG SECTION C
                                ;       CAUTION: DO NOT SET BIT 14 DURING THIS TEST. IT WILL LOCK
                                ;       UP THE BUS
                                ;
000614                          ST2PC::
000614  012746  000220                MOV     #<PLRWR+PLC>,-(SP)
000620                                CALL    CPCR                    ;DIRECT CNTL WORD TO CP
000624  016746  000000G               MOV     BINWD,-(SP)             ;MOVE DATA WORD TO LOD BUS REG
000630                                CALL    LBCP                    ;SEND DATA TO CP
000634  005046                        CLR     -(SP)
000636                                CALL    CPCR                    ; CLEAR THE CONTROL REGISTER    NS
000642  000167  000252                JMP     ST2LP                   ;TEST FOR REPEAT
                                ;
                                ;       CP PIPELINE REG SECTION D
                                ;       CAUTION: DO NOT SET BIT 11 DURING THIS TEST. IT WILL LOCK
                                ;       UP THE BUS
                                ;
000646                          ST2PD::
000646  012746  000230                MOV     #<PLRWR+PLD>,-(SP)
000652                                CALL    CPCR                    ;DIRECT CNTL WORD TO CP
000656  016746  000000G               MOV     BINWD,-(SP)             ;MOVE DATA WORD TO LOD BUS REG
000662                                CALL    LBCP                    ;SEND DATA TO CP
000666  005046                        CLR     -(SP)                   ;                               NS
000670                                CALL    CPCR                    ; CLEAR THE CONTROL REGISTER    NS
000674  000167  000220                JMP     ST2LP
                                ;
                                ;       WRITE DMA ADDRESS REGISTER
                                ;
000700                          ST2DA::
000700  012746  000210                MOV     #<PLRWR+PLB>,-(SP)
000704                                CALL    CPCR                    ;SET UP PLR B
000710  012746  000005                MOV     #DMAAWRH,-(SP)          ;SET VALUE FOR CONTROL WORD
000714                                CALL    LBCP                    ;SEND WORD TO CP
000720  016746  000000G               MOV     BINWD,-(SP)             ;SET VALUE FOR DMA REG
000724                                CALL    LBCP                    ;TRANSFER TO CP
000730  005046                        CLR     -(SP)                   ;CLEAR CP CONTROL REG
000732                                CALL    CPCR
000736  000167  000156                JMP     ST2LP
                                ;
                                ;       WRITE DMA DATA REGISTER
                                ;
000742                          ST2DD::
000742  012746  000210                MOV     #<PLRWR+PLB>,-(SP)
000746                                CALL    CPCR                    ;SET UP PLR B
000752  012746  000004                MOV     #DMARWRH,-(SP)          ;SET VALUE FOR CONTROL WORD
000756                                CALL    LBCP                    ;SEND WORD TO CP
000762  016746  000000G               MOV     BINWD,-(SP)             ;SET VALUE FOR DMA REG
000766                                CALL    LBCP                    ;TRANSFER TO CP
000772  005046                        CLR     -(SP)                   ;CLEAR CP CONTROL REG
000774                                CALL    CPCR
001000  000167  000114                JMP     ST2LP
                                ;
                                ;       LOAD CP REGISTERS
                                ;
```

```
 286 001004  005267  000000G·        ST2RF:: INC·     INDEX·
 287 001010  005267  000000G·        ST2RE:: INC·     INDEX·
 288 001014  005267  000000G·        ST2RD:: INC·     INDEX·
 289 001020  005267  000000G·        ST2RC:: INC·     INDEX·
 290 001024  005267  000000G·        ST2RB:: INC·     INDEX·
 291 001030  005267  000000G·        ST2RA:: INC·     INDEX·
 292·001034  005267  000000G·        ST2R9:: INC·     INDEX·
 293 001040  005267  000000G·        ST2R8:: INC·     INDEX·
 294 001044  005267  000000G·        ST2R7:: INC·     INDEX·
 295 001050  005267  000000G·        ST2R6:: INC·     INDEX·
 296 001054  005267  000000G·        ST2R5:: INC·     INDEX·
 297 001060  005267  000000G·        ST2R4:: INC·     INDEX·
 298 001064  005267  000000G·        ST2R3:: INC·     INDEX·
 299 001070  005267  000000G·        ST2R2:: INC·     INDEX·
 300 001074  005267  000000G·        ST2R1:: INC·     INDEX·
 301 001100                          ST2R0::
 302·                                ;
 303 001100  016700  000000G·                MO·      INDEX,R0                ;PREPARE·TO·CALL·SUBRTN·
 304 001104                                  CALL·    LDREG·                  ;LOAD·REGISTER·
 305 001110  005067  000000G·                CLR·     INDEX·
 306 001114  000167  000000                  JMP·     ST2LP·
 307                                 ;
 308                                 ;
 309                                 ;
 310 001120                          ST2LP:
 311 001120  032767  000000G·000000G·        BIT·     #LOOP,BASE·             ;REPEAT·?·
 312·001126  001402·                         BEQ·     ST2X                    ;NO·
 313 001130  000167  177234                  JMP·     ST2IN·
 314                                 ;
 315 001134                          ST2X:
 316 001134                                  CALL·    KILL                    ;KILL·AST·
 317 001140  000167  000000G·                JMP·     CPXX
```

```
                                ;
                                ;
                                ;       READ·
                                ;       PERFORM·THIRD·LEVEL·PARSING·
                                ;       EG. IN·THE·COMMAND:
                                ;       CP>RE·CR·
                                ;       PARSE·THE·'CR'
                                ;
                                ;
001144                          RE2::
001144                                  CALL·   FIND                    ;FIND·A·REG·MNEMONIC IN·COMMAND·LINE·
001150  103004                          BCC·    1$                      ;OK, CONTINUE·
001152·                                 CALL·   ERR4                    ;NOTHING·THERE·
001156  000167  000346                  JMP·    RE2X                    ;RETURN·TO TOP·OF·LOOP·(PROMPT)
                                ;
                                ;       MATCH·THE·REGISTER·MNEMONIC·FROM THE·COMMAND·LINE·AGAINST·
                                ;       THE·TABLE·OF VALID·MNEMONICS.
                                ;
001162· 012700  000023          1$:     MOV·    #RE2LN,R0               ;NUMBER·OF TABLE·ENTRIES·
001166  012702· 000144'                 MOV·    #RE2TBL,R2·             ;R2·-> TABLE·
001172·                                 CALL·   SCAN                    ;MATCH·AGAINST·COMMAND·LINE·
001176  103004                          BCC·    2$                      ;OK, CONTINUE·
001200                                  CALL·   ERR6
001204  000167  000320                  JMP·    RE2X
                                ;
                                ;       SAVE·THE·POINTER·TO·THE·ROUTINE·ASSOCIATED WITH·THE·
                                ;       REGISTER, R1 -> ROUTINE·ADDRESS,
                                ;       CALL·ROUTINE TO·SCAN·COMMAND·LINE·FOR·LOOP INDICATOR.
                                ;       EG. CP>RE·CR L·
                                ;       LOOP·FLAG·WILL·BE·SET·IF·INDICATOR·IS·PRESENT,
                                ;       JUMP·TO·ROUTINE·TO·LOAD·REGISTER.
                                ;
001210  010167  000000G·        2$:     MOV·    R1,RTNPT·               ;SAVE·POINTER·TO·RTN
001214                                  CALL·   LOOPR·                  ;LOOP?·
001220  016701  000000G·        RE2IN:  MOV·    RTNPT,R1                ;POINT·TO·ROUTINE·
001224  000171  000000                  JMP·    @(R1)                   ;EXECUTE·ROUTINE·
                                ;
                                ;
                                ;       MEMORY·ADDRESS·REG·
                                ;
001230                          RE2MA::
001230  012746  000002                  MOV·    #<MAROUT>,-(SP)
001234                                  CALL·   CPCR                    ;DIRECT·CNTL·WORD·TO CP·
001240                                  CALL·   CPLB                    ;REQUEST·CP·TO·LOD·BUS·
001244  005046                          CLR·    -(SP)                   ; CLEAR·THE·CONTROL·REGISTER·    NS·
001246                                  CALL·   CPCR                    ;                                NS·
001252· 012601                          MOV·    (SP)+,R1                ;CP·WORD·RETURNED·ON STACK·
001254  000167  000202                  JMP·    RE2PUT·
                                ;
                                ;       MICROPGM·ADDRESS·REG·
                                ;
001260                          RE2MP::
001260  012746  000004                  MOV·    #CSADRD,-(SP)
001264                                  CALL·   CPCR                    ;DIRECT·CNTL·WORD·TO CP·
001270                                  CALL·   CPLB                    ;REQUEST·CP·TO·LOD·BUS·
001274  005046                          CLR·    -(SP)                   ; CLEAR·THE·CONTROL·REGISTER·    NS·
001276                                  CALL·   CPCR
```

```
376 001302  012601                        MOV     (SP)+,R1                ;CP WORD RETURNED ON STACK
377 001304  042701  176000                BIC     #176000,R1              ;CLEAR USELESS BITS
378 001310  000167  000146                JMP     RE2PUT
379                                 ;
380                                 ;       READ DMA DATA REGISTER
381                                 ;
382 001314                          RE2DD::
383 001314  012746  000210                MOV     #<PLRWR+PLB>,-(SP)
384 001320                                CALL    CPCR                    ;DIRECT CNTL WORD TO CP
385 001324  012746  000003                MOV     #DMARRDH,-(SP)          ;SET VALUE FOR CONTROL WORD
386 001330                                CALL    LBCP                    ;SEND DATA TO CP
387 001334                                CALL    CPLB                    ;READ FROM CP
388 001340  012601                        MOV     (SP)+,R1                ;LOAD VALUE INTO R1 FOR PRINT
389 001342  000167  000114                JMP     RE2PUT
390                                 ;
391                                 ;       PRINT CP REGISTERS
392                                 ;
393 001346  005267  000000G       RE2RF:: INC     INDEX
394 001352  005267  000000G       RE2RE:: INC     INDEX
395 001356  005267  000000G       RE2RD:: INC     INDEX
396 001362  005267  000000G       RE2RC:: INC     INDEX
397 001366  005267  000000G       RE2RB:: INC     INDEX
398 001372  005267  000000G       RE2RA:: INC     INDEX
399 001376  005267  000000G       RE2R9:: INC     INDEX
400 001402  005267  000000G       RE2R8:: INC     INDEX
401 001406  005267  000000G       RE2R7:: INC     INDEX
402 001412  005267  000000G       RE2R6:: INC     INDEX
403 001416  005267  000000G       RE2R5:: INC     INDEX
404 001422  005267  000000G       RE2R4:: INC     INDEX
405 001426  005267  000000G       RE2R3:: INC     INDEX
406 001432  005267  000000G       RE2R2:: INC     INDEX
407 001436  005267  000000G       RE2R1:: INC     INDEX
408 001442                          RE2R0::
409                                 ;
410 001442  016700  000000G               MOV     INDEX,R0                ;PREPARE TO CALL SUBRTN
411 001446                                CALL    REREG                   ;READ A REGISTER
412 001452  005067  000000G               CLR     INDEX
413 001456  000167  000000                JMP     RE2PUT
414                                 ;
415                                 ;
416 001462  032767  000000G 000000G RE2PUT: BIT     #ONCE,BASE              ;PRINTED ONCE ?
417 001470  001011                        BNE     1$                      ;YES, SKIP
418 001472  052767  000000G 000000G       BIS     #ONCE,BASE              ;SET FLAG FOR PRINTED ONCE
419 001500  012705  000000G               MOV     #PRINT,R5               ;POINT TO PRINT LINE
420 001504                                CALL    UNPK                    ;CONVERT VALUE IN R1 FOR PRINTING
421 001510                                CALL    CONSOL                  ;PRINT ON CONSOLE
422                                 ;
423 001514  032767  000000G 000000G 1$:   BIT     #LOOP,BASE              ;REPEAT
424 001522  001402                        BEQ     RE2X                    ;NO, EXIT
425 001524  000167  177470                JMP     RE2IN                   ;EXECUTE ROUTINE AGAIN
426                                 ;
427 001530                          RE2X:
428 001530  042767  000000G 000000G       BIC     #ONCE,BASE              ;CLEAR PRINT CONTROL FLAG
429 001536                                CALL    KILL
430 001542  000167  000000G               JMP     CPXX
```

```
                                        ;
                                        ;
                                        ;       RESET·CP·
                                        ;
                                        ;
001546                                  RS2::
001546  005046                                  CLR·    -(SP)                   ;RESET·NOTHING·                 FW·
001550  012746  000002                          MOV·    #Q$CSET,-(SP)           ;SET·CP·RESET·                  FW·
001554                                          CALL·   CSR1                    ;SEND·                          FW·
001560  012746  000002                          MOV·    #Q$CSET,-(SP)           ;CLEAR·CP·RESET·                FW·
001564  005046                                  CLR·    -(SP)                   ;SET·NOTHING·                   FW·
001566                                          CALL·   CSR1                    ;DO·IT·                         FW·
001572· 000167  000000G·                        JMP·    CPXX
                                        ;
                                        ;
                                        ;
                                        ;       CALL·HOR·LOADER·
                                        ;
                                        ;
001576                                  CL2::
001576                                          CALL·   CL·                     ;CALL·ROUTINE·IN·'MAIN'
001602· 000167  000000G·                        JMP·    CPXX
                                        ;
        000001                                  .END·
```

```
ALUCKE= 040000
ALUOE = 004000
A01   = 010000
BASE  = ****** GX
BINWD = ****** GX
BITVAL= 000000
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BYTE0 = 000000
BYTE1 = 000001
BYTE10= 000012
BYTE11= 000013
BYTE12= 000014
BYTE13= 000015
BYTE14= 000016
BYTE15= 000017
BYTE16= 000020
BYTE17= 000021
BYTE18= 000022
BYTE19= 000023
BYTE2 = 000002
BYTE20= 000024
BYTE21= 000025
BYTE22= 000026
BYTE23= 000027
BYTE24= 000030
BYTE25= 000031
BYTE26= 000032
BYTE27= 000033
BYTE28= 000034
BYTE29= 000035
BYTE3 = 000003
BYTE30= 000036
BYTE31= 000037
BYTE32= 000040
BYTE33= 000041
BYTE34= 000042
BYTE35= 000043
BYTE36= 000044
BYTE37= 000045
BYTE38= 000046
BYTE39= 000047
BYTE4 = 000004
BYTE40= 000050
BYTE41= 000051
BYTE42= 000052
BYTE43= 000053
BYTE44= 000054
BYTE45= 000055
BYTE46= 000056
BYTE47= 000057
BYTE48= 000060
BYTE49= 000061
BYTE5 = 000005
BYTE50= 000062
BYTE51= 000063
BYTE52= 000064
BYTE53= 000065
BYTE54= 000066
BYTE55= 000067
BYTE56= 000070
BYTE57= 000071
BYTE58= 000072
BYTE59= 000073
BYTE6 = 000006
BYTE60= 000074
BYTE61= 000075
BYTE62= 000076
BYTE63= 000077
BYTE64= 000100
BYTE65= 000101
BYTE66= 000102
BYTE67= 000103
BYTE68= 000104
BYTE69= 000105
BYTE7 = 000007
BYTE70= 000106
BYTE71= 000107
BYTE72= 000110
BYTE73= 000111
BYTE74= 000112
BYTE75= 000113
BYTE76= 000114
BYTE77= 000115
BYTE78= 000116
BYTE79= 000117
BYTE8 = 000010
BYTE80= 000120
BYTE81= 000121
BYTE82= 000122
BYTE83= 000123
BYTE84= 000124
BYTE85= 000125
BYTE86= 000126
BYTE87= 000127
BYTE88= 000130
BYTE89= 000131
BYTE9 = 000011
BYTE90= 000132
BYTE91= 000133
BYTE92= 000134
BYTE93= 000135
BYTE94= 000136
BYTE95= 000137
BYTE96= 000140
BYTE97= 000141
BYTE98= 000142
BYTE99= 000143
BYTVAL= 000144
CBKALL= 001000
CBKCLK= 000400
CL    = ****** GX
CL2   = 001576RG    002
CNOBRE= 100000
CONSOL= ****** GX
CPCCEN= 010000
CPCR  = ****** GX
CPLB  = ****** GX
CPREAD= 040000
CPWRTE= 020000
CPXX  = ****** GX
CSADRD= 000004
CSEQCI= 100000
CSOE  = 000040
CSR1  = ****** GX
CSWRTE= 000100
DBR.RD= 000001
DB$CPP= 001457
DB$SPT= 000026
DE$TPC= 000023
DISPGS= 100000
DMAAWR= 000005
DMARRD= 000003
DMARWR= 000004
ENBR  = 010000
ERR4  = ****** GX
ERR5  = ****** GX
ERR6  = ****** GX
FIND  = ****** GX
INDEX = ****** GX
KILL  = ****** GX
LBCP  = ****** GX
LDREG = ****** GX
LOC.EN= 000100
LOC.WA= 040000
LOC.WB= 100000
LOOP  = ****** GX
LOOPR = ****** GX
MAREN1= 000001
MAREN2= 004000
MARLOD= 010000
MAROUT= 000002
MAR.LO= 002000
MAR.OU= 000040
MBKALL= 001000
MBKCLK= 000400
MMADRD= 000100
MMLEFT= 000002
MMOE  = 000004
MMWRTE= 000010
MNOBRE= 100000
MREN1 = 000001
MREN2 = 020000
MSYN  = 000040
N     = 000144
ONCE  = ****** GX
PACK  = ****** GX
PLB   = 000010
PLC   = 000020
PLD   = 000030
PLRWR = 000200
PLR.EN= 000200
PRINT = ****** GX
QR$CR1= 176420
QR$CR2= 176422
QR$LBR= 176424
Q$ATTN= 000100
Q$BCL = 000001
Q$CCCP= 000040
Q$CHB = 000400
Q$CHRL= 000200
Q$CLR = 000040
Q$CNC = 030000
Q$CP  = 000060
Q$CPCC= 000010
Q$CP2 = 000260
Q$CSC = 010000
Q$CSEL= 000360
Q$CSET= 000002
Q$CSP = 020000
Q$DMA = 000001
Q$ENBK= 040000
Q$ENOP= 020000
Q$FAL = 004000
Q$FC  = 000045
Q$FO  = 000044
Q$FP  = 000046
Q$HBF = 000002
Q$ICP = 000006
Q$IHB = 000003
Q$IHRL= 000002
Q$IMRP= 000007
Q$LBD = 001000
Q$LBDP= 001001
Q$LBP = 000001
Q$LDCD= 000003
Q$LDMD= 000004
Q$LDPP= 002000
Q$LHP = 010000
Q$MNC = 140000
Q$MP  = 000050
Q$MRP = 000040
Q$MRP2= 000240
Q$MSC = 040000
Q$MSET= 000004
Q$MSP = 100000
Q$NCLK= 176000
Q$PP  = 000100
Q$PPSW= 000320
Q$PP2 = 000300
Q$QHLT= 000013
Q$QL  = 000043
Q$QLA = 000053
Q$QLB = 000054
Q$QLR = 000001
Q$QW  = 000042
Q$RDCD= 000005
Q$RDMD= 000006
Q$REBK= 001000
Q$RNC = 006000
Q$RSC = 004000
Q$RSET= 000010
Q$SM  = 100000
Q$SP  = 000120
Q$SP2 = 000340
REREG = ****** GX
RE2   = 001144RG    002
RE2DD = 001314RG    002
RE2IN = 001220R     002
RE2LN = 000023 G
RE2MA = 001230RG    002
RE2MP = 001260RG    002
RE2PUT= 001462R     002
RE2RA = 001372RG    002
RE2RB = 001366RG    002
RE2RC = 001362RG    002
RE2RD = 001356RG    002
RE2RE = 001352RG    002
RE2RF = 001346RG    002
RE2R0 = 001442RG    002
RE2R1 = 001436RG    002
RE2R2 = 001432RG    002
RE2R3 = 001426RG    002
RE2R4 = 001422RG    002
RE2R5 = 001416RG    002
RE2R6 = 001412RG    002
RE2R7 = 001406RG    002
RE2R8 = 001402RG    002
RE2R9 = 001376RG    002
RE2TBL= 000144RG    002
RE2X  = 001530R     002
RGQ.EN= 000200
RGQ.VA= 020000
RS2   = 001546RG    002
RTNPT = ****** GX
SCAN  = ****** GX
SEQ.CI= 000010
ST2    = 000260RG   002
ST2BK  = 000414RG   002
ST2CR  = 000400RG   002
ST2DA  = 000700RG   002
```

```
CPREST  M[illegible]D M1110  27-MAR-80 14:44  PAGE 7-2
SYMBOL T[illegible]
```

ST2DD	000742RG	002	TD$OAR	176344	T$SC	000022	WORD33	000102	WORD68	000210
ST2IN	000370R	002	TD$OTR	176346	T$SCLK	020000	WORD34	000104	WORD69	000212
ST2LN	000031 G		TD$QRD	000274	T$SEG1	000000	WORD35	000106	WORD7	000016
ST2LP	001120R	002	TD$SW	176376	T$SEG2	000001	WORD36	000110	WORD70	000214
ST2MA	000440RG	002	TD$TAR	176372	T$SEG3	000002	WORD37	000112	WORD71	000216
ST2PA	000530RG	002	TD$TAW	176362	T$SO	000001	WORD38	000114	WORD72	000220
ST2PB	000562RG	002	TD$TDR	176374	T$UBUS	100000	WORD39	000116	WORD73	000222
ST2PC	000614RG	002	TD$TDW	176364	T$1CLK	000400	WORD4	000010	WORD74	000224
ST2PD	000646RG	002	T$AD	000020	T$8BEN	000020	WORD40	000120	WORD75	000226
ST2RA	001030RG	002	T$BA	000002	UBD.IN	000020	WORD41	000122	WORD76	000230
ST2RB	001024RG	002	T$BD	000010	UNPK	****** GX	WORD42	000124	WORD77	000232
ST2RC	001020RG	002	T$BSO	100000	WORD0	000000	WORD43	000126	WORD78	000234
ST2RD	001014RG	002	T$BT	000020	WORD1	000002	WORD44	000130	WORD79	000236
ST2RE	001010RG	002	T$BTAR	000030	WORD10	000024	WORD45	000132	WORD8	000020
ST2RF	001004RG	002	T$BTD	002000	WORD11	000026	WORD46	000134	WORD80	000240
ST2R0	001100RG	002	T$CD	000100	WORD12	000030	WORD47	000136	WORD81	000242
ST2R1	001074RG	002	T$CLK	002000	WORD13	000032	WORD48	000140	WORD82	000244
ST2R2	001070RG	002	T$DISK	000200	WORD14	000034	WORD49	000142	WORD83	000246
ST2R3	001064RG	002	T$DRD	000004	WORD15	000036	WORD5	000012	WORD84	000250
ST2R4	001060RG	002	T$EMEM	010000	WORD16	000040	WORD50	000144	WORD85	000252
ST2R5	001054RG	002	T$FSAA	000000	WORD17	000042	WORD51	000146	WORD86	000254
ST2R6	001050RG	002	T$FSAB	000004	WORD18	000044	WORD52	000150	WORD87	000256
ST2R7	001044RG	002	T$FSAC	000014	WORD19	000046	WORD53	000152	WORD88	000260
ST2R8	001040RG	002	T$FSB2	000010	WORD2	000004	WORD54	000154	WORD89	000262
ST2R9	001034RG	002	T$IB	000026	WORD20	000050	WORD55	000156	WORD9	000022
ST2TBL	000000RG	002	T$IBAR	000024	WORD21	000052	WORD56	000160	WORD90	000264
ST2X	001134R	002	T$IBE	020000	WORD22	000054	WORD57	000162	WORD91	000266
S$CLR	000000		T$IBF	040000	WORD23	000056	WORD58	000164	WORD92	000270
S$LA	000001		T$ICD	000040	WORD24	000060	WORD59	000166	WORD93	000272
S$QB	000005		T$MODE	004000	WORD25	000062	WORD6	000014	WORD94	000274
S$QR	000006		T$OB	000036	WORD26	000064	WORD60	000170	WORD95	000276
S$QX	000004		T$OBE	004000	WORD27	000066	WORD61	000172	WORD96	000300
S$SR	000007		T$OBF	010000	WORD28	000070	WORD62	000174	WORD97	000302
S$S1	000010		T$OBRA	000034	WORD29	000072	WORD63	000176	WORD98	000304
S$S2	000014		T$OBWA	000032	WORD3	000006	WORD64	000200	WORD99	000306
TD$CTR	176370		T$OUTA	100000	WORD30	000074	WORD65	000202	WRDVAL	000310
TD$CTW	176360		T$RBDO	000200	WORD31	000076	WORD66	000204	XTREAD	001000
TD$INL	004000		T$RNB	000040	WORD32	000100	WORD67	000206	XTWRTE	000400
TD$MEM	000270		T$RSET	040000						

```
. ABS.  000000     000
        000000     001
CPREST  001606     002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3426 WORDS  ( 14 PAGES)
DYNAMIC MEMORY:  4916 WORDS  ( 18 PAGES)
ELAPSED TIME:  00:00:50
CPREST,CPREST/-SP=[20,1]IM,[20,1]CPREST
```

```
                                        .TITLE GRLD
000000                                  .PSECT CPLD
                                ;
                                ;
                                ;       HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
                                ;       CONTROL PROCESSOR TEST ROUTINES
                                ;       *******   PROTOTYPE VERSION   *******
                                ;       MEMORY LOAD ROUTINES
                                ;
                                ;       COMMANDS:
                                ;       LD      LOAD CP MEMORIES
                                ;
                                ;       ONCE A COMMAND HAS BEEN EXECUTED (OR AN ERROR ENCOUNTERED)
                                ;       THIS MODULE RETURNS CONTROL TO THE MODULE MRP AT LOCATION
                                ;       'CPXX'.
                                ;
                                ;
                                ; **********************
                                ;
                                ;       NOTE:
                                ;       THE COMMAND TO LOAD CP DATA MEMORY CANNOT BE USED
                                ;       UNLESS MRP MICROPROGRAM MEMORY AND CP CONTROL STORE
                                ;       HAVE BEEN LOADED VIA THE 'LOADER' PROGRAM.
                                ;
                                ;       IF THE QEX MEMORIES ARE EITHER LOADED OR PRINTED AFTER
                                ;       CP DATA MEMORY HAS BEEN LOADED, CP DATA MEMORY CONTENTS
                                ;       ARE LOST.
                                ;
                                ; **********************
                                ;
                                ;
                                        .MCALL  WTSE$S,CLEF$S
                                ;
                                ;
                                ;       TABLE OF VALID CP MEMORY MNEMONICS AND ADDRESSES OF
                                ;       ASSOCIATED ROUTINES
                                ;
000000                          LD2TBL::
000000    103     123                   .ASCII  /CS/                    ;LOAD MICROPGM MEMORY
000002  000272'                         .WORD   LD2CS
000004    103     104                   .ASCII  /CD/                    ;LOAD DATA MEMORY
000006  000472'                         .WORD   LD2CD
000010    110     114                   .ASCII  /HL/                    ;LOAD HRL BUFFER
000012  000734'                         .WORD   LD2HL
000014    102     114                   .ASCII  /BL/                    ;LOAD BCL
000016  001114'                         .WORD   LD2BL
        000004                  LD2LN   ==      <.-LD2TBL>/4
                                ;
                                ;
                                ;       LOAD
                                ;       PERFORM THIRD LEVEL PARSING
                                ;       EG. IN THE COMMAND:
                                ;       CP>LD CD 0
                                ;       PARSE TH 'CD'
                                ;
                                ;
000020                          LD2::
```

```
 58 000020                                          CALL    FIND                    ;LOCATE THE MEMORY MNEMONIC IN THE COMMAND LINE.
 59 000024  103004                                  BCC     1$                      ;OK, CONTINUE.
 60 000026                                          CALL    ERR4                    ;'MISSING OPERAND'
 61 000032  000167  001236                          JMP     LD2X                    ;EXIT.
 62                                         ;
 63                                         ;       MATCH MNEMONIC IN THE COMMAND LINE AGAINST TABLE OF VALID MNEMONICS.
 64                                         ;
 65 000036  012700  000004          1$:     MOV     #LD2LN,R0               ;NUMBER OF TABLE ENTRIES.
 66 000042  012702  000000'                 MOV     #LD2TBL,R2              ;POINT TO TABLE.
 67 000046                                          CALL    SCAN                    ;MATCH AGAINST COMMAND LINE.
 68 000052  103004                                  BCC     2$                      ;MATCH WAS MADE.
 69 000054                                          CALL    ERR7                    ;'INVALID MEMORY MNEMONIC'
 70 000060  000167  001210                          JMP     LD2X
 71                                         ;
 72                                         ;       SAVE POINTER TO ROUTINE ASSOCIATED WITH THE MEMORY MNEMONIC,
 73                                         ;       SCAN THE REMAINDER OF THE COMMAND LINE FOR CONTROL INFORMATION:
 74                                         ;
 75                                         ;       1. START ADDRESS ONLY; REPEAT PROMPT FOR EACH MEMORY LOCATION
 76                                         ;          FROM THE START ADDRESS FORWARD. NB. END OF MEMORY (AS DETECTED
 77                                         ;          BY THE QMAIN SUBROUTINE PDATA) OR A <CR> RESPONSE TO THE PROMPT
 78                                         ;          WILL TERMINATE THE LOAD.
 79                                         ;          CP>LD CD 0
 80                                         ;       2. START ADDRESS, LOOP INDICATOR. LOOP ON THE LOADING OF
 81                                         ;          THIS ONE MEMORY LOCATION ONLY.
 82                                         ;          CP>LD CD 0 L
 83                                         ;       3. START ADDRESS, END ADDRESS, NO LOOP; ISSUE ONLY ONE
 84                                         ;          PROMPT AND FILL MEMORY (BETWEEN START AND END ADDRESSES)
 85                                         ;          WITH THIS VALUE.
 86                                         ;          CP>LD CD 0 7
 87                                         ;       4. START ADDRESS, END ADDRESS, LOOP INDICATOR; ISSUE ONLY ONE
 88                                         ;          PROMPT AND LOOP ON THE LOADING OF MEMORY (BETWEEN START
 89                                         ;          AND END ADDRESSES) WITH THIS VALUE.
 90                                         ;          CP>LD CD 0 7 L
 91                                         ;
 92                                         ;
 93 000064  010167  000000G·        2$:     MOV     R1,RTNPT                ;SAVE POINTER
 94 000070                                          CALL    FIND                    ;LOCATE START ADDRESS IN COMMAND LINE.
 95 000074  103004                                  BCC     3$                      ;OK, CONTINUE.
 96 000076                                          CALL    ERR4                    ;'MISSING OPERAND'
 97 000102  000167  001166                          JMP     LD2X                    ;EXIT.
 98 000106                                  3$:     CALL    PACK                    ;CONVERT COMMAND LINE VALUE TO BINARY.
 99 000112  103004                                  BCC     4$                      ;CONVERSION SUCCESSFUL.
100 000114                                          CALL    ERR5                    ;'INVALID NUMERIC VALUE'
101 000120  000167  001150                          JMP     LD2X
102                                        ;
103 000124  016767  000000G·000000G·4$:    MOV     BINWD,MSTRT             ;SAVE LOADING START ADDRESS.
104 000132  016767  000000G·000000G·       MOV     BINWD,MSTR2             ;SAVE IT TWICE (FOR REFRESH ON LOOP)
105 000140  012767  177777  000000G·       MOV     #-1,MEND                ;INIT END ADDRESS.
106                                        ;
107                                        ;       START ADDRESS HAS BEEN FOUND. SCAN FOR END ADDRESS OR LOOP
108                                        ;       INDICATOR (CONDITIONS 2, 3 ABOVE). IF THERE IS NOTHING
109                                        ;       FURTHER IN THE COMMAND LINE, CONDITION 1 IS IN EFFECT.
110                                        ;
111 000146                                         CALL    FIND                    ;SCAN COMMAND LINE.
112 000152  103004                                 BCC     5$                      ;SOMETHING THERE
113 000154  052767  000000G·000000G·              BIS     #RP,BASE                ;SIGNAL TO REPEAT PROMPT
114 000162  000437                                 BR      9$                      ;JUMP TO RTN.
```

```
                                        ;
000164  122711  000114          5$:    CMPB·   #'L,(R1)                ;LOOP·INDICATOR·
000170  001006                         BNE·    6$                      ;NO, MUST·BE·UPPER·ADDRESS·
000172· 016767  000000G·000000G·        MOV·    MSTRT,MEND·             ;SET·END·ADDR·=·START·ADDR·
000200                                  CALL·   HANG                    ;HOW·TO·STOP·LOOP·
000204  000426                         BR      9$                      ;JUMP·TO·RTN·
                                        ;
000206                          6$:     CALL·   PACK                    ;CONVERT·UPPER·ADDRESS·
000212· 103004                          BCC·    7$                      ;OK, CONTINUE·
000214                                  CALL·   ERR5                    ;INVALID·NUMERIC·
000220  000167  001050                  JMP·    LD2X                    ;EXIT·
                                        ;
                                        ;       SAVE·END·ADDRESS·(BINARY)
                                        ;       CHECK·FOR·LOOP·INDICATOR·AFTER·END·ADDRESS (CONDITION·4)
                                        ;
000224  016767  000000G·000000G·7$:     MOV·    BINWD,MEND·             ;SET·UP·END·ADDRESS·
000232·                                 CALL·   FIND                    ;CHECK·FOR·LOOP·INDICATOR·
000236  103411                          BCS·    9$                      ;NO·LOOP·
000240  122711  000114                  CMPB·   #'L,(R1)                ;CORRECT·INDICATOR·
000244  001404                          BEQ·    8$                      ;YES, CONTINUE·
000246                                  CALL·   ERR11                   ;LOOP·OPTION·ERROR·
000252· 000167  001016                  JMP·    LD2X
000256                          8$:     CALL·   HANG                    ;HOW·TO·STOP·LOOP·
                                        ;
000262· 016701  000000G·        9$:     MOV·    RTNPT,R1                ;POINT·TO·ROUTINE·
000266  000171  000000·                 JMP·    @(R1)                   ;JUMP·TO·ROUTINE·
                                        ;
                                        ;
                                        ;       LOAD·CONTROL STORE·
                                        ;
                                        ;
000272·                         LD2CS::
000272· 005046                          CLR·    -(SP)                   ;CLEAR·NOTHING·
000274  012746  000002                  MOV·    #Q$CSET,-(SP)           ;CP·RESET·
000300                                  CALL·   CSR1                    ;DO·IT·
000304  012746  000002                  MOV·    #Q$CSET,-(SP)           ;CLEAR·CP·RESET·
000310  005046                          CLR·    -(SP)                   ;SET·NOTHING·
000312·                                 CALL·   CSR1
                                        ;
000316  016746  000000G·                MOV·    CSHIGH,-(SP)            ;SUPPLY·UPPER·MEMORY LIMIT·
000322· 016746  000000G·                MOV·    CSLOW,-(SP)             ;LOWER·LIMIT·
000326                                  CALL·   BUFS4                   ;PREPARE·TO·LOAD·
000332· 103002·                         BCC·    100$                    ;NO·ERRORS, CONTINUE
000334  000167  000734                  JMP·    LD2X
                                        ;
000340  032767  000000G·000000G·100$:   BIT·    #RP,BASE·               ;REPEAT·PROMPT·
000346  001416                          BEQ·    2$                      ;NO, ONCE·ONLY·
000350                          1$:     CALL·   PDATA·                  ;READ·DATA FROM·COMMAND·LINE·
000354  103002·                         BCC·    10$                     ;NOT·END·OF·MEMORY·
000356  000167  000712                  JMP·    LD2X                    ;IF·END, EXIT·
000362· 102002·                 10$:    BVC·    20$                     ;<CR> RESPONSE·TO·PROMPT·
000364  000167  000704                  JMP·    LD2X                    ;EXIT·ON·<CR> RESPONSE·
000370                          20$:    CALL·   LOADCS·                 ;LOAD·4 SECTIONS·
000374  066767  000000G·000000G·        ADD·    INCVAL,MSTR2·           ;BUMP·LOAD ADDRESS·
000402· 000762·                         BR      1$                      ;REPEAT·
                                        ;
                                        ;       PROMPT·ONCE,·THEN·FILL·MEMORY·
```

```
                                        ;
000404                          2$:     CALL    PDATA                   ;PROMPT
000410  103002                         BCC     30$                     ;NOT END OF MEMORY
000412  000167  000656                  JMP     LD2X                    ;ELSE EXIT
000416  102002                 30$:     BVC     3$                      ;NOT <CR> RESPONSE
000420  000167  000650                  JMP     LD2X
                                        ;
000424                          3$:     CALL    LOADCS                  ;LOAD 4 SECTIONS
000430  066767  000000G 000000G         ADD     INCVAL,MSTR2            ;ADVANCE ADDRESS
000436  026767  000000G 000000G         CMP     MSTR2,MEND              ;HAS UPPER MEMORY LIMIT BEEN REACHED
000444  101767                          BLOS    3$                      ;NO, CONTINUE
000446  032767  000000G 000000G         BIT     #LOOP,BASE              ;LOOP ON ?
000454  001002                          BNE     4$                      ;YES, CONTINUE
000456  000167  000612                  JMP     LD2X                    ;NO LOOP, EXIT
000462  016767  000000G 000000G 4$:     MOV     MSTRT,MSTR2             ;REINITIALIZE ADDRESS
000470  000755                          BR      3$
                                        ;
                                        ;
                                        ;       LOAD DATA MEMORY
                                        ;
000472                          LD2CD::
000472  016746  000000G                 MOV     CDHIGH,-(SP)            ;SUPPLY UPPER MEMORY LIMIT
000476  016746  000000G                 MOV     CDLOW,-(SP)             ;LOWER LIMIT
000502                                  CALL    BUFSET                  ;PREPARE FOR LOAD
000506  103002                         BCC     100$                    ;NO ERROR, CONTINUE
000510  000167  000560                  JMP     LD2X
                                        ;
000514  005046                  100$:   CLR     -(SP)                   ;START MICROCODE AT 0
000516                                  CALL    SEQCS
                                        ;
000522  005046                          CLR     -(SP)                   ;REINHIBIT BRANCH CONTROL REGISTER
000524                                  CALL    CPCR
                                        ;
000530  012746  000377                  MOV     #377,-(SP)              ;SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
000534                                  CALL    SEQMM
                                        ;
000540  005046                          CLR     -(SP)                   ;REINHIBIT BRANCH CONTROL REGISTER
000542                                  CALL    MRPCR
                                        ;
000546  012767  001000  176422          MOV     #Q$REBK,QR$CR2          ;RE-ARM INTERRUPTS
000554  012767  120000  176422          MOV     #<Q$SM+Q$ENOP>,QR$CR2   ;SET SEARCH MODE + ENABLE INTERRUPTS
000562  012746  000360                  MOV     #Q$CSEL,-(SP)           ;CLEAR ALL SELECTIONS
000566  052716  001001                  BIS     #<Q$LBD+Q$LBP>,(SP)     ;CLEAR DRIVE AND PULSE
000572  052716  030000                  BIS     #Q$CNC,(SP)             ;CLEAR CP NO-CLOCK
000576  005046                          CLR     -(SP)                   ;SET NOTHING
000600                                  CALL    CSR1
                                        ;
000604  032767  000000G 000000G         BIT     #RP,BASE                ;REPEAT PROMPT ?
000612  001412                          BEQ     2$                      ;NO
000614                          1$:     CALL    PDATA                   ;READ DATA FROM COMMAND LINE
000620  103434                          BCS     LDCDX
000622  102433                          BVS     LDCDX                   ;<CR> RESPONSE
000624                                  CALL    LOADCD                  ;LOAD ONE WORD
000630  066767  000000G 000000G         ADD     INCVAL,MSTR2            ;BUMP LOAD ADDRESS
000636  000766                          BR      1$                      ;REPEAT
                                        ;
                                        ;       PROMPT ONCE THEN FILL MEMORY
```

```
                                        ;
000640                          2$:     CALL    PDATA           ;PROMPT
000644  103422                         BCS     LDCDX           ;END OF MEMORY
000646  102421                         BVS     LDCDX           ;<CR> RESPONSE
000650                          3$:     CALL    LOADCD          ;LOAD 1 WORD
000654  066767  000000G 000000G        ADD     INCVAL,MSTR2    ;ADVANCE ADDRESS
000662  026767  000000G 000000G        CMP     MSTR2,MEND      ;HAS UPPER MEMORY LIMIT BEEN REACHED
000670  101767                         BLOS    3$              ;NO, CONTINUE
000672  032767  000000G 000000G        BIT     #LOOP,BASE      ;LOOP ON ?
000700  001404                         BEQ     LDCDX           ;NO, EXIT
000702  016767  000000G 000000G        MOV     MSTRT,MSTR2     ;REINITIALIZE ADDRESS
000710  000757                         BR      3$
                                        ;
000712  005046                  LDCDX:  CLR     -(SP)           ;CLEAR NOTHING IN CSR1
000714  012746  176000                 MOV     #0$NCLK,-(SP)   ;SET NO-CLOCKS
000720                                  CALL    CSR1
000724  005067  176422                 CLR     @R$CR2          ;SET LOAD MODE
000730  000167  000340                 JMP     LD2X            ;EXIT LOAD
                                        ;
                                        ;       LOAD HRL BUFFER IN MODULE 'CPBUG'
                                        ;
000734                          LD2HL::
000734  016746  000000G                 MOV     HLHIGH,-(SP)    ;SUPPLY UPPER MEMORY LIMIT
000740  016746  000000G                 MOV     HLLOW,-(SP)     ;SUPPLY LOWER MEMORY LIMIT
000744                                  CALL    BUFSM           ;VERIFY CURRENT RANGE
000750  103551                         BCS     LD2X            ;AN ERROR
                                        ;
000752  032767  000000G 000000G        BIT     #RP,BASE        ;REPEAT PROMPT
000760  001421                         BEQ     3$              ;NO, FILL WITH ONE VALUE
000762                          1$:     CALL    PDATA           ;PROMPT FOR DATA
000766  103002                         BCC     2$              ;NOT END OF MEMORY
000770  000167  000300                 JMP     LD2X            ;ERROR OR END OF MEMORY
000774  102002                  2$:     BVC     20$             ;NOT <CR> RESPONSE
000776  000167  000272                 JMP     LD2X
                                        ;
001002  016700  000000G         20$:    MOV     MSTR2,R0        ;LOAD MEMORY ADDR (REALLY AN OFFSET)
001006  016760  000000G 000000G        MOV     DATA1,HRL(R0)   ;LOAD HRL TABLE
001014  066767  000000G 000000G        ADD     INCVAL,MSTR2    ;BUMP OFFSET
001022  000757                         BR      1$              ;PROMPT FOR NEXT VALUE
                                        ;
                                        ;       PROMPT ONCE THEN FILL MEMORY
                                        ;
001024                          3$:     CALL    PDATA           ;PROMPT FOR VALUE
001030  103002                         BCC     4$              ;NOT END OF MEMORY
001032  000167  000236                 JMP     LD2X            ;ERROR ON PROMPT
001036  102002                  4$:     BVC     40$             ;NOT <CR> RESPONSE
001040  000167  000230                 JMP     LD2X
001044  016700  000000G         40$:    MOV     MSTR2,R0        ;LOAD ADDR (REALLY AN OFFSET)
001050  016760  000000G 000000G        MOV     DATA1,HRL(R0)   ;MOVE VALUE TO HRL IN MEMORY
001056  066767  000000G 000000G        ADD     INCVAL,MSTR2    ;BUMP TO NEXT ADDRESS
001064  026767  000000G 000000G        CMP     MSTR2,MEND      ;FINISHED ?
001072  101761                         BLOS    4$              ;NO, FILL NEXT LOCATION
001074  032767  000000G 000000G        BIT     #LOOP,BASE      ;REPEAT ?
001102  001474                         BEQ     LD2X            ;NO, EXIT
001104  016767  000000G 000000G        MOV     MSTRT,MSTR2     ;RE-INIT START ADDRESS (OFFSET)
001112  000751                         BR      4$
                                        ;
```

```
                                ;
                                ;       LOAD BCL BUFFER IN MODULE 'CPBUG'
                                ;
001114                          LD2BL::
001114  016746  000000G         MOV     BLHIGH,-(SP)            ;SUPPLY UPPER MEMORY LIMIT
001120  016746  000000G         MOV     BLLOW,-(SP)             ;SUPPLY LOWER MEMORY LIMIT
001124                          CALL    BUFSM                   ;VERIFY CURRENT RANGE
001130  103461                  BCS     LD2X                    ;AN ERROR
                                ;
001132  032767  000000G 000000G BIT     #RP,BASE                ;REPEAT PROMPT
001140  001421                  BEQ     3$                      ;NO, FILL WITH ONE VALUE
001142                          1$:     CALL    PDATA           ;PROMPT FOR DATA
001146  103002                  BCC     2$                      ;NOT END OF MEMORY
001150  000167  000120          JMP     LD2X                    ;ERROR OR END OF MEMORY
001154  102002                  2$:     BVC     20$             ;NOT <CR> RESPONSE
001156  000167  000112          JMP     LD2X
                                ;
001162  016700  000000G         20$:    MOV     MSTR2,R0        ;LOAD MEMORY ADDR (REALLY AN OFFSET)
001166  016760  000000G 000000G         MOV     DATA1,BCL(R0)   ;LOAD BCL TABLE
001174  066767  000000G 000000G         ADD     INCVAL,MSTR2    ;BUMP OFFSET
001202  000757                  BR      1$                      ;PROMPT FOR NEXT VALUE
                                ;
                                ;       PROMPT ONCE THEN FILL MEMORY
                                ;
001204                          3$:     CALL    PDATA           ;PROMPT FOR VALUE
001210  103002                  BCC     4$                      ;NOT END OF MEMORY
001212  000167  000056          JMP     LD2X                    ;ERROR ON PROMPT
001216  102002                  4$:     BVC     40$             ;NOT <CR> RESPONSE
001220  000167  000050          JMP     LD2X
                                ;
001224  016700  000000G         40$:    MOV     MSTR2,R0        ;LOAD ADDR (REALLY AN OFFSET)
001230  016760  000000G 000000G         MOV     DATA1,BCL(R0)   ;MOVE VALUE TO BCL IN MEMORY
001236  066767  000000G 000000G         ADD     INCVAL,MSTR2    ;BUMP TO NEXT ADDRESS
001244  026767  000000G 000000G         CMP     MSTR2,MEND      ;FINISHED ?
001252  101761                  BLOS    4$                      ;NO, FILL NEXT LOCATION
001254  032767  000000G 000000G         BIT     #LOOP,BASE      ;REPEAT ?
001262  001404                  BEQ     LD2X                    ;NO, EXIT
001264  016767  000000G 000000G         MOV     MSTRT,MSTR2     ;RE-INIT START ADDRESS (OFFSET)
001272  000751                  BR      4$
                                ;
                                ;
001274                          LD2X:
001274  012746  000130          MOV     #<CSWRTEN+PLD>,-(SP)    ;WRITE DISABL SECT D    FW
001300  005046                  CLR     -(SP)                   ;SET NOTHING                    FW
001302                          CALL    CSR1                    ;                               FW
001306  042767  000000G 000000G BIC     #RP,BASE                ;CLEAR PROMPT REPEAT FLAG
001314                          CALL    KILL                    ;KILL AST
001320  000167  000000G         JMP     CPXX
```

```
                                        ;
                                        ;
                                        ;       LOAD·CP·CONTROL·STORE·
                                        ;
                                        ;
001324                                  LOADCS:
001324  316746  000000G·                       MOV·    MSTR2,-(SP)             ;SAVE·CURRENT·ADDRESS·IN·STACK
001330                                          CALL·   SEQCS·                  ;SEQUENCE·UP·TO·START·ADDRESS·
001334  012746  000100                          MOV·    #CSWRTEN,-(SP)          ;SET·WRITE·ENABLE·SECT·A·
001340                                          CALL·   CPCRA·                  ;WRITE·CP·CONTROL·REG·
001344  016746  000000G·                       MOV·    DATA1,-(SP)             ;SUPPLY·SECT·A·DATA·WORD·
001350                                          CALL·   LBCSC·                  ;SEND·IT·TO·CP·-·CLOCK·SEQUENCER·ONLY·
                                        ;
001354  016746  000000G·                       MOV·    MSTR2,-(SP)             ;SAVE·CURRENT·ADDRESS·IN·STACK
001360                                          CALL·   SEQCS·                  ;SEQUENCE·UP·TO·START·ADDRESS·
001364  012746  000110                          MOV·    #<CSWRTEN+PLB>,-(SP)    ;SET·WRITE·ENABLE·SECT·B·
001370                                          CALL·   CPCRA·                  ;DIRECT·CNTL·WORD·TO·CP·
001374  016746  000000G·                       MOV·    DATA2,-(SP)             ;MOVE·DATA·WORD·TO·LOD·BUS·REG
001400                                          CALL·   LBCSC·                  ;SEND·DATA·TO·CP·-·CLOCK·SEQUENCER·ONLY·
                                        ;
001404  016746  000000G·                       MOV·    MSTR2,-(SP)             ;SAVE·CURRENT·ADDRESS·IN·STACK
001410                                          CALL·   SEQCS·                  ;SEQUENCE·UP·TO·START·ADDRESS·
001414  012746  000120                          MOV·    #<CSWRTEN+PLC>,-(SP)    ;SET·WRITE·ENABLE·SECT·C·
001420                                          CALL·   CPCRA·                  ;DIRECT·CNTL·WORD·TO·CP·
001424  016746  000000G·                       MOV·    DATA3,-(SP)             ;MOVE·DATA·WORD·TO·LOD·BUS·REG
001430                                          CALL·   LBCSC·                  ;SEND·DATA·TO·CP·-·CLOCK·SEQUENCER·ONLY·
                                        ;
001434  016746  000000G·                       MOV·    MSTR2,-(SP)             ;SAVE·CURRENT·ADDRESS·IN·STACK
001440                                          CALL·   SEQCS·                  ;SEQUENCE·UP·TO·START·ADDRESS·
001444  012746  000130                          MOV·    #<CSWRTEN+PLD>,-(SP)    ;WRITE·ENABLE·SECTION·D·
001450                                          CALL·   CPCRA·                  ;DIRECT·CNTL·WORD·TO·CP·
001454  016746  000000G·                       MOV·    DATA4,-(SP)             ;MOVE·DATA·WORD·TO·LOD·BUS·REG
001460                                          CALL·   LBCSC·                  ;SEND·DATA·TO·CP·-·CLOCK·SEQUENCER·ONLY·
                                        ;
001464  005046                                 CLR·    -(SP)                   ;REINHIBIT·BRANCH·CONTROL·REG·
001466                                          CALL·   CPCR
001472·005046                                   CLR·    -(SP)
001474                                          CALL·   LBCP                    ;SINGLE·CLOCK·TO·REINHIBIT·BRANCH·REGISTER·
001500                                          RETURN·
                                        ;
                                        ;
                                        ;       LOAD·CP·DATA MEMORY·
                                        ;
                                        ;
001502·                                 LOADCD:
001502·012767  000003  176424                   MOV·    #Q$LDCD,QR$LBR          ;MOVE·ATTN·CODE·TO·LOD·BUS·REG
001510  012767  120100  176422·                 MOV·    #<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2·;SET·ATTN·CODE·READY
001516·016701  176422                   1$:     MOV·    QR$CR2,R1               ;READ·CSR2
001522·032701  000100                           BIT·    #Q$ATTN,R1              ;ATTN·CLEAR·
001526  001373                                  BNE·    1$                      ;NO, READ·AGAIN·
                                        ;
001530  016767  000000G·176424                  MOV·    MSTR2,QR$LBR·           ;CD·MEMORY·START·ADDRESS·
001536  012767  120040  176422·                 MOV·    #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2·;SET·CC·TO·CP·
001544  016701  176422                   2$:     MOV·    QR$CR2,R1               ;READ·CSR2
001550  032701  000040                           BIT·    #Q$CCCP,R1              ;IS·CC·TO·CP·CLEAR·
001554  001373                                  BNE·    2$                      ;NO, READ·AGAIN·
                                        ;
```

```
392 001556  012767  000001  176424          MOV     #1,@R$LBR               ;TRANSFER COUNT = 1 WORD
393 001564  012767  120040  176422          MOV     #<Q$CCCP+Q$SM+Q$ENOP>,@R$CR2 ;SET CC TO CP
394 001572  016701  176422          3$:     MOV     @R$CR2,R1               ;READ CSR2
395 001576  032701  000040                  BIT     #Q$CCCP,R1              ;IS CC TO CP CLEAR
396 001602  001373                          BNE     3$                      ;NO, READ AGAIN
397                                 ;
398 001604  012767  000000G 176424          MOV     #DATA1,@R$LBR           ;CC MEMORY DATA BUFFER
399 001612  012767  120040  176422          MOV     #<Q$CCCP+Q$SM+Q$ENOP>,@R$CR2 ;SET CC TO CP
400                                 ;
401                                 ;       WAIT FOR INTERRUPT FROM CP
402                                 ;
403 001620                                  WTSE$S  #EFN.3
404                                 ;
405 001632                                  CLEF$S  #EFN.3
406                                 ;
407                                 ;       RE-ARM INTERRUPTS
408                                 ;
409 001644  012767  100400  176422          MOV     #<Q$SM+Q$CHB>,@R$CR2    ;CLEAR INTERRUPT (USE HIT BUFFER INT)
410 001652  012767  101000  176422          MOV     #<Q$SM+Q$REBK>,@R$CR2   ;RE-ARM
411 001660  012767  160000  176422          MOV     #<Q$SM+Q$ENBK+Q$ENOP>,@R$CR2 ;ENABLE
412 001666                                  RETURN
413                                 ;
414         000001                          .END
```

```
ALUCKE= 040000
ALUOE = 004000
A01   = 010000
BASE  = ****** G
BCL   = ****** GX
BINWD = ****** GX
BITVAL= 000000
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BLHIGH= ****** GX
BLLOW = ****** GX
BUFSET= ****** GX
BUFSM = ****** GX
BUFS4 = ****** GX
BYTE0 = 000000
BYTE1 = 000001
BYTE10= 000012
BYTE11= 000013
BYTE12= 000014
BYTE13= 000015
BYTE14= 000016
BYTE15= 000017
BYTE16= 000020
BYTE17= 000021
BYTE18= 000022
BYTE19= 000023
BYTE2 = 000002
BYTE20= 000024
BYTE21= 000025
BYTE22= 000026
BYTE23= 000027
BYTE24= 000030
BYTE25= 000031
BYTE26= 000032
BYTE27= 000033
BYTE28= 000034
BYTE29= 000035
BYTE3 = 000003
BYTE30= 000036
BYTE31= 000037
BYTE32= 000040
BYTE33= 000041
BYTE34= 000042
BYTE35= 000043
BYTE36= 000044
BYTE37= 000045
BYTE38= 000046
BYTE39= 000047
BYTE4 = 000004
BYTE40= 000050
BYTE41= 000051
BYTE42= 000052
BYTE43= 000053
BYTE44= 000054
BYTE45= 000055
BYTE46= 000056
BYTE47= 000057
BYTE48= 000060
BYTE49= 000061
BYTE5 = 000005
BYTE50= 000062
BYTE51= 000063
BYTE52= 000064
BYTE53= 000065
BYTE54= 000066
BYTE55= 000067
BYTE56= 000070
BYTE57= 000071
BYTE58= 000072
BYTE59= 000073
BYTE6 = 000006
BYTE60= 000074
BYTE61= 000075
BYTE62= 000076
BYTE63= 000077
BYTE64= 000100
BYTE65= 000101
BYTE66= 000102
BYTE67= 000103
BYTE68= 000104
BYTE69= 000105
BYTE7 = 000007
BYTE70= 000106
BYTE71= 000107
BYTE72= 000110
BYTE73= 000111
BYTE74= 000112
BYTE75= 000113
BYTE76= 000114
BYTE77= 000115
BYTE78= 000116
BYTE79= 000117
BYTE8 = 000010
BYTE80= 000120
BYTE81= 000121
BYTE82= 000122
BYTE83= 000123
BYTE84= 000124
BYTE85= 000125
BYTE86= 000126
BYTE87= 000127
BYTE88= 000130
BYTE89= 000131
BYTE9 = 000011
BYTE90= 000132
BYTE91= 000133
BYTE92= 000134
BYTE93= 000135
BYTE94= 000136
BYTE95= 000137
BYTE96= 000140
BYTE97= 000141
BYTE98= 000142
BYTE99= 000143
BYTVAL= 000144
CBKALL= 001000
CBKCLK= 000400
CDHIGH= ****** GX
CDLOW = ****** GX
CNOBRE= 100000
CPCCEN= 010000
CPCR  = ****** GX
CPCRA = ****** GX
CPREAD= 040000
CPWRTE= 020000
CPXX  = ****** GX
CSADRD= 000004
CSEOCI= 100000
CSHIGH= ****** GX
CSLOW = ****** GX
CSOE  = 000040
CSR1  = ****** GX
CSWRTE= 000100
DATA1 = ****** GX
DATA2 = ****** GX
DATA3 = ****** GX
DATA4 = ****** GX
DBR.RD= 000001
DB$CPP= 001457
DB$SPT= 000026
DB$TPC= 000023
DISPGS= 100000
DMAAWR= 000005
DMARRD= 000003
DMARWR= 000004
EFN.3 = ****** GX
ENBR  = 010000
ERR11 = ****** GX
ERR4  = ****** GX
ERR5  = ****** GX
ERR7  = ****** GX
FIND  = ****** GX
HANG  = ****** GX
HLHIGH= ****** GX
HLLOW = ****** GX
HRL   = ****** GX
INCVAL= ****** GX
KILL  = ****** GX
LBCP  = ****** GX
LBCSC = ****** GX
LDCDX   000712R
LD2     000020RG
LD2BL   001114RG
LD2CD   000472RG
LD2CS   000272RG
LD2HL   000734RG
LD2LN = 000004 G
LD2TBL  000000RG
LD2X    001274R
LOADCD  001502R
LOADCS  001324R
LOC.EN= 000100
LOC.WA= 040000
LOC.WB= 100000
LOOP  = ****** GX
MAREN1= 000001
MAREN2= 004000
MARLOD= 010000
MAROUT= 000002
MAR.LO= 002000
MAR.OU= 000040
MBKALL= 001000
MBKCLK= 000400
MEND  = ****** GX
MMADRD= 000100
MMLEFT= 000002
MMOE  = 000004
MMWRTE= 000010
MNOBRE= 100000
MREN1 = 000001
MREN2 = 020000
MRPCR = ****** GX
MSTRT = ****** GX
MSTR2 = ****** GX
MSYN  = 000040
N     = 000144
PACK  = ****** GX
PDATA = ****** GX
PLB   = 000010
PLC   = 000020
PLD   = 000030
PLRWR = 000200
PLR.EN= 000200
QR$CR1= 176420
QR$CR2= 176422
QR$LBR= 176424
Q$ATTN= 000100
Q$BCL = 000001
Q$CCCP= 000040
Q$CHB = 000400
Q$CHRL= 000200
Q$CLR = 000040
Q$CNC = 030000
Q$CP  = 000060
Q$CPCC= 000010
Q$CP2 = 000260
Q$CSC = 010000
002 Q$CSEL= 000360
002 Q$CSET= 000002
002 Q$CSP = 020000
002 Q$DMA = 000001
002 Q$ENBK= 040000
002 Q$ENOP= 020000
Q$FAL = 004000
002 Q$FC  = 000045
002 Q$FO  = 000044
002 Q$FP  = 000046
002 Q$HBF = 000002
Q$ICP = 000006
Q$IHB = 000003
Q$IHRL= 000002
Q$IMRP= 000007
Q$LBD = 001000
Q$LBDP= 001001
Q$LBP = 000001
Q$LDCD= 000003
Q$LDMD= 000004
Q$LDPP= 002000
Q$LHP = 010000
Q$MNC = 140000
Q$MR  = 000052
Q$MRP = 000040
Q$MRP2= 000240
Q$MSC = 040000
Q$MSET= 000004
Q$MSP = 100000
Q$NCLK= 176000
Q$PP  = 000100
Q$PPSW= 000320
Q$PP2 = 000300
Q$QHLT= 000013
Q$QL  = 000043
Q$QLA = 000053
Q$QLB = 000054
Q$QLR = 000001
Q$QW  = 000042
Q$RDCD= 000005
Q$RDMD= 000006
Q$REBK= 001000
Q$RNC = 006000
Q$RSC = 004000
Q$RSET= 000010
Q$SM  = 100000
Q$SP  = 000120
Q$SP2 = 000340
RGQ.EN= 000200
RGQ.VA= 020000
RP    = ****** GX
RTNPT = ****** GX
SCAN  = ****** GX
SEQCS = ****** GX
```

```
CPLD  MACRO M1110  27-MAR-80 14:43  PAGE 6-3
SYMBOL TABLE
```

```
SEQMM = ****** GX      T$FSAA = 000000      WORD12 = 000030      WORD42 = 000124      WORD72 = 000220
SEQ.CI = 000010        T$FSAB = 000004      WORD13 = 000032      WORD43 = 000126      WORD73 = 000222
S$CLR = 000000         T$FSAC = 000014      WORD14 = 000034      WORD44 = 000130      WORD74 = 000224
S$LA  = 000001         T$FSB2 = 000010      WORD15 = 000036      WORD45 = 000132      WORD75 = 000226
S$QB  = 000005         T$IB   = 000026      WORD16 = 000040      WORD46 = 000134      WORD76 = 000230
S$QR  = 000006         T$IBAR = 000024      WORD17 = 000042      WORD47 = 000136      WORD77 = 000232
S$QX  = 000004         T$IBE  = 020000      WORD18 = 000044      WORD48 = 000140      WORD78 = 000234
S$SR  = 000007         T$IBF  = 040000      WORD19 = 000046      WORD49 = 000142      WORD79 = 000236
S$S1  = 000010         T$ICD  = 000040      WORD2  = 000004      WORD5  = 000012      WORD8  = 000020
S$S2  = 000014         T$MODE = 004000      WORD20 = 000050      WORD50 = 000144      WORD80 = 000240
TD$CTR = 176370        T$OB   = 000036      WORD21 = 000052      WORD51 = 000146      WORD81 = 000242
TD$CTW = 176360        T$OBE  = 004000      WORD22 = 000054      WORD52 = 000150      WORD82 = 000244
TD$INL = 004000        T$OBF  = 010000      WORD23 = 000056      WORD53 = 000152      WORD83 = 000246
TD$MEM = 000270        T$OBRA = 000034      WORD24 = 000060      WORD54 = 000154      WORD84 = 000250
TD$OAR = 176344        T$OBWA = 000032      WORD25 = 000062      WORD55 = 000156      WORD85 = 000252
TD$OTR = 176346        T$OUTA = 100000      WORD26 = 000064      WORD56 = 000160      WORD86 = 000254
TD$ORD = 000274        T$RBDO = 000200      WORD27 = 000066      WORD57 = 000162      WORD87 = 000256
TD$SW  = 176376        T$RNB  = 000040      WORD28 = 000070      WORD58 = 000164      WORD88 = 000260
TD$TAR = 176372        T$RSET = 040000      WORD29 = 000072      WORD59 = 000166      WORD89 = 000262
TD$TAW = 176362        T$SC   = 000022      WORD3  = 000006      WORD6  = 000014      WORD9  = 000022
TD$TDR = 176374        T$SCLK = 020000      WORD30 = 000074      WORD60 = 000170      WORD90 = 000264
TD$TDW = 176364        T$SEG1 = 000000      WORD31 = 000076      WORD61 = 000172      WORD91 = 000266
T$AD   = 000020        T$SEG2 = 000001      WORD32 = 000100      WORD62 = 000174      WORD92 = 000270
T$BA   = 000002        T$SEG3 = 000002      WORD33 = 000102      WORD63 = 000176      WORD93 = 000272
T$BD   = 000010        T$SO   = 000001      WORD34 = 000104      WORD64 = 000200      WORD94 = 000274
T$BSO  = 100000        T$UBUS = 100000      WORD35 = 000106      WORD65 = 000202      WORD95 = 000276
T$BT   = 000020        T$1CLK = 000400      WORD36 = 000110      WORD66 = 000204      WORD96 = 000300
T$BTAR = 000030        T$8BEN = 000020      WORD37 = 000112      WORD67 = 000206      WORD97 = 000302
T$BTD  = 002000        UBD.IN = 000020      WORD38 = 000114      WORD68 = 000210      WORD98 = 000304
T$CD   = 000100        WORD0  = 000000      WORD39 = 000116      WORD69 = 000212      WORD99 = 000306
T$CLK  = 002000        WORD1  = 000002      WORD4  = 000010      WORD7  = 000016      WRDVAL = 000310
T$DISK = 000200        WORD10 = 000024      WORD40 = 000120      WORD70 = 000214      XTREAD = 001000
T$DRD  = 000004        WORD11 = 000026      WORD41 = 000122      WORD71 = 000216      XTWRTE = 000400
T$EMEM = 010000
```

```
. ABS.   000000     000
         000000     001
CPLD     001670     002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3677 WORDS  ( 15 PAGES)
DYNAMIC MEMORY:  4916 WORDS  ( 18 PAGES)
ELAPSED TIME:  00:00:51
CPLD,CPLD/-SP=[20,1]IM,[20,1]CPLD
```

```
                                .TITLE GRPR
000000                          .PSECT CPPR
                        ;
                        ;
                        ;       HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
                        ;       CONTROL PROCESSOR TEST ROUTINES.
                        ;       *******   PROTOTYPE VERSION   *******
                        ;
                        ;       COMMANDS:
                        ;       PR      PRINT CP MEMORY CONTENTS
                        ;
                        ;       ONCE A COMMAND HAS BEEN EXECUTED (OR AN ERROR ENCOUNTERED)
                        ;       THIS MODULE RETURNS CONTROL TO THE MODULE CP AT LOCATION
                        ;       'CPXX'.
                        ;
                        ;
                        ; ********************
                        ;
                        ;       NOTE:
                        ;       THE COMMAND TO PRINT CP DATA MEMORY CANNOT BE USED
                        ;       UNLESS MRP MICROPROGRAM MEMORY AND CP CONTROL STORE
                        ;       HAVE BEEN LOADED VIA THE 'LOADER' PROGRAM.
                        ;
                        ; ***********************
                        ;
                        ;
                                .MCALL  WTSE$S,CLEF$S
                        ;
                        ;
                        ;       TABLE OF VALID CP MEMORY MNEMONICS AND ADDRESSES OF ASSOCIATED
                        ;       ROUTINES
                        ;
000000                  PR2TBL::
000000  103     123             .ASCII  /CS/                    ;PRINT MICROPGM MEMORY
000002  000312'                 .WORD   PR2CS
000004  103     104             .ASCII  /CD/                    ;PRINT DATA MEMORY
000006  000632'                 .WORD   PR2CD
        000002          PR2LN   ==      <.-PR2TBL>/4
                        ;
                        ;       PRINT BUFFER
                        ;
000010                  PB2TBL::
000010  110     102             .ASCII  /HB/                    ;PRINT HIT BUFFER
000012  001314'                 .WORD   PB2HB
000014  110     114             .ASCII  /HL/                    ;PRINT HRL
000016  001406'                 .WORD   PB2HL
000020  102     114             .ASCII  /BL/                    ;PRINT BCL
000022  001500'                 .WORD   PB2BL
        000003          PB2LN   ==      <.-PB2TBL>/4
                        ;
                        ;
                        ;       PRINT
                        ;       PERFORM THIRD LEVEL PARSING
                        ;       EG. IN THE COMMAND:
                        ;       PR>PR CD 0
                        ;       PARSE THE 'CD'
                        ;
```

```
                                        ;
                                PR2::
                                        CALL    FIND                    ;LOCATE MEMORY MNEMONIC IN COMMAND LINE.
                                        BCC     1$                      ;OK, CONTINUE.
                                        CALL    ERR4                    ;'MISSING OPERAND'
                                        JMP     PR2X                    ;EXIT.
                                        ;
                                1$:     MOV     #PR2LN,R0               ;NUMBER OF TABLE ENTRIES.
                                        MOV     #PR2TBL,R2              ;POINT TO TABLE.
                                        CALL    SCAN                    ;MATCH AGAINST COMMAND LINE.
                                        BCC     2$                      ;MATCH WAS MADE.
                                        CALL    ERR7                    ;'INVALID MEMORY MNEMONIC'
                                        JMP     PR2X
                                        ;
                                        ;       SAVE POINTER TO ROUTINE ASSOCIATED WITH THE MEMORY MNEMONIC,
                                        ;       SCAN THE REMAINDER OF THE COMMAND LINE FOR CONTROL INFORMATION:
                                        ;
                                        ;       1. START ADDRESS ONLY, PRINT ONE MEMORY LOCATION.
                                        ;          CP>PR CD 0
                                        ;       2. START ADDRESS, LOOP INDICATOR, LOOP ON THE READING OF
                                        ;          THIS ONE MEMORY LOCATION ONLY, PRINT THE CONTENTS ONLY
                                        ;          ONCE.
                                        ;          CP>LD CD 0 L
                                        ;       3. START ADDRESS, END ADDRESS, NO LOOP, PRINT THE CONTENTS
                                        ;          OF MEMORY BETWEEN THE START AND END ADDRESSES.
                                        ;          CP>PR CD 0 7
                                        ;       4. START ADDRESS, END ADDRESS, LOOP INDICATOR, READ THE CONTENTS
                                        ;          OF MEMORY BETWEEN THE START AND END ADDRESSES, HOWEVER, PRINT
                                        ;          ONLY THE CONTENTS OF MEMORY AT THE START ADDRESS.
                                        ;          CP>PR CD 0 7 L
                                        ;
                                        ;
000070  010167  000000G         2$:     MOV     R1,RTNPT                ;SAVE POINTER.
000074                                  CALL    FIND                    ;LOCATE NON-BLANK IN COMMAND LINE.
000100  103004                          BCC     3$                      ;OK, CONTINUE.
000102                                  CALL    ERR4                    ;'MISSING OPERAND'
000106  000167  001114                  JMP     PR2X                    ;EXIT.
000112                          3$:     CALL    PACK                    ;CONVERT COMMAND LINE VALUE TO BINARY.
000116  103004                          BCC     4$                      ;CONVERSION SUCCESSFUL.
000120                                  CALL    ERR5                    ;'INVALID NUMERIC VALUE'
000124  000167  001076                  JMP     PR2X
                                        ;
000130  016767  000000G 000000G 4$:     MOV     BINWD,MSTRT             ;SAVE PRINT START ADDRESS.
000136  016767  000000G 000000G         MOV     BINWD,MSTR2             ;SAVE IT TWICE.
000144  012767  177777  000000G         MOV     #-1,MEND                ;INIT END ADDRESS.
                                        ;
                                        ;       START ADDRESS HAS BEEN FOUND, SCAN FOR END ADDRESS OR
                                        ;       LOOP INDICATOR (CONDITIONS 2, 3 ABOVE). IF THERE IS NOTHING
                                        ;       FURTHER IN THE COMMAND LINE, CONDITION 1 IS IN EFFECT.
                                        ;
000152                                  CALL    FIND                    ;SCAN COMMAND LINE.
000156  103004                          BCC     5$                      ;SOMETHING THERE.
000160  016767  000000G 000000G         MOV     MSTRT,MEND              ;SET END ADDR = START ADDR.
000166  000445                          BR       9$                     ;JUMP TO RTN.
                                        ;
000170  122711  000114          5$:     CMPB    #'L,(R1)                ;LOOP INDICATOR.
000174  001011                          BNE     6$                      ;NO, MUST BE UPPER ADDRESS.
```

```
115 000176  052767  000000G 000000G          BIS     #OUT,BASE               ;SET FLAG FOR OUTPUT CONTROL
116 000204  016767  000000G 000000G          MOV     MSTRT,MEND              ;SET END ADDR = START ADDR
117 000212                                  CALL    HANG                    ;HOW TO STOP LOOP
118 000216  000431                          BR      9$                      ;JUMP TO RTN
119                                 ;
120 000220                          6$:     CALL    PACK                    ;CONVERT UPPER ADDRESS
121 000224  103004                          BCC     7$                      ;OK, CONTINUE
122 000226                                  CALL    ERR5                    ;INVALID NUMERIC
123 000232  000167  000770                  JMP     PR2X                    ;EXIT
124                                 ;
125                                 ;       SAVE END ADDRESS (BINARY)
126                                 ;       CHECK FOR LOOP INDICATOR AFTER END ADDRESS (CONDITION 4)
127                                 ;
128 000236  016767  000000G 000000G 7$:     MOV     BINWD,MEND              ;SET UP END ADDRESS
129 000244                                  CALL    FIND                    ;CHECK FOR LOOP INDICATOR
130 000250  103414                          BCS     9$                      ;NO LOOP
131 000252  122711  000114                  CMPB    #'L,(R1)                ;CORRECT LOOP INDICATOR
132 000256  001404                          BEQ     8$                      ;YES
133 000260                                  CALL    ERR11
134 000264  000167  000736                  JMP     PR2X                    ;NO
135 000270  052767  000000G 000000G 8$:     BIS     #OUT,BASE               ;SET OUTPUT CONTROL
136 000276                                  CALL    HANG                    ;HOW TO STOP LOOP
137                                 ;
138 000302  016701  000000G         9$:     MOV     RTNPT,R1                ;POINT TO ROUTINE
139 000306  000171  000000                  JMP     @(R1)                   ;JUMP TO ROUTINE
140                                 ;
141                                 ;
142                                 ;       PRINT FROM CONTROL STORE
143                                 ;
144                                 ;
145 000312                          PR2CS::
146 000312  005046                          CLR     -(SP)                   ;CLEAR NOTHING
147 000314  012746  000002                  MOV     #Q$CSET,-(SP)           ;CP RESET
148 000320                                  CALL    CSR1                    ;DO IT
149 000324  012746  000002                  MOV     #Q$CSET,-(SP)           ;CLEAR CP RESET
150 000330  005046                          CLR     -(SP)                   ;SET NOTHING
151 000332                                  CALL    CSR1
152                                 ;
153 000336  016746  000000G                 MOV     CSHIGH,-(SP)            ;SUPPLY UPPER MEMORY LIMIT
154 000342  016746  000000G                 MOV     CSLOW,-(SP)             ;LOWER LIMIT
155 000346                                  CALL    BUFS4                   ;PREPARE FOR LOAD
156 000352  103002                          BCC     1$                      ;NO ERROR
157 000354  000167  000646                  JMP     PR2X
158                                 ;
159 000360  016746  000000G         1$:     MOV     MSTR2,-(SP)             ;SEND CURRENT ADDRESS
160 000364                                  CALL    SEQCS                   ;SEQUENCE UP TO PRINT START ADDRESS
161 000370  012746  000040                  MOV     #CSOE,-(SP)             ;SELECT CNTL STORE SECT A
162 000374                                  CALL    CPCR                    ;SET CP CNTL REG
163 000400  005046                          CLR     -(SP)
164 000402                                  CALL    LBCP                    ;CLOCK TO RESET BR REG FF
165 000406                                  CALL    CPLB                    ;REQUEST CP TO LOD BUS
166 000412  012667  000000G                 MOV     (SP)+,DATA1             ;SAVE FOR PRINTING
167                                 ;
168 000416  016746  000000G                 MOV     MSTR2,-(SP)             ;SEND CURRENT ADDRESS
169 000422                                  CALL    SEQCS                   ;SEQUENCE UP TO PRINT START ADDRESS
170 000426  012746  000050                  MOV     #<CSOE+PLB>,-(SP)       ;SELECT CNTL STORE SECT B
171 000432                                  CALL    CPCR                    ;SET CP CNTL REG
```

```
172 000436  005046                        CLR     -(SP)
173 000440                                  CALL    LBCP                ;CLOCK TO REINHIBIT BR
174 000444                                  CALL    CPLB                ;REQUEST CP TO LOD BUS
175 000450  012667  000000G                 MOV     (SP)+,DATA2         ;SAVE FOR PRINTING
176                             ;
177 000454  016746  000000G                 MOV     MSTR2,-(SP)         ;SEND CURRENT ADDRESS
178 000460                                  CALL    SEQCS               ;SEQUENCE UP TO PRINT START ADDRESS
179 000464  012746  000060                  MOV     #<CSOE+PLC>,-(SP)   ;SELECT CNTL STORE SECT C
180 000470                                  CALL    CPCR                ;SET CP CNTL REG
181 000474  005046                          CLR     -(SP)
182 000476                                  CALL    LBCP                ;CLOCK TO REINHIBIT BR
183 000502                                  CALL    CPLB                ;REQUEST CP TO LOD BUS
184 000506  012667  000000G                 MOV     (SP)+,DATA3         ;SAVE FOR PRINTING
185                             ;
186 000512  016746  000000G                 MOV     MSTR2,-(SP)         ;SEND CURRENT ADDRESS
187 000516                                  CALL    SEQCS               ;SEQUENCE UP TO PRINT START ADDRESS
188 000522  012746  000070                  MOV     #<CSOE+PLD>,-(SP)   ;SELECT CNTL STORE SECT D
189 000526                                  CALL    CPCR                ;SET CP CNTL REG
190 000532  005046                          CLR     -(SP)
191 000534                                  CALL    LBCP                ;CLOCK TO REINHIBIT BR
192 000540                                  CALL    CPLB                ;REQUEST CP TO LOD BUS
193 000544  012667  000000G                 MOV     (SP)+,DATA4         ;SAVE FOR PRINTING
194 000550  005046                          CLR     -(SP)               ;CLEAR CP CONTROL REG
195 000552                                  CALL    CPCR
196                             ;
197 000556                                  CALL    PRDATA              ;PRINT MEMORY CONTENTS
198 000562  103002                          BCC     100$                ;NOT END OF MEMORY
199 000564  000167  000436                  JMP     PR2X                ;END OF MEMORY
200                             ;
201 000570  066767  000000G 000000G 100$:   ADD     INCVAL,MSTR2        ;ADVANCE ADDRESS
202 000576  026767  000000G 000000G         CMP     MSTR2,MEND          ;HAS UPPER MEMORY LIMIT BEEN REACHED
203 000604  101665                          BLOS    1$                  ;NO, CONTINUE
204 000606  032767  000000G 000000G         BIT     #LOOP,BASE          ;LOOP ON ?
205 000614  001002                          BNE     200$                ;YES, CONTINUE
206 000616  000167  000404                  JMP     PR2X                ;NO, EXIT
207 000622  016767  000000G 000000G 200$:   MOV     MSTRT,MSTR2         ;INIT START ADDRESS
208 000630  000653                          BR      1$                  ;AND REPEAT PRINT
209                             ;
210                             ;
211                             ;       PRINT FROM DATA MEMORY
212                             ;
213 000632                      PR2CD::
214 000632  016746  000000G                 MOV     CDHIGH,-(SP)        ;SUPPLY MEMORY UPPER LIMIT
215 000636  016746  000000G                 MOV     CDLOW,-(SP)         ;LOWER LIMIT
216 000642                                  CALL    BUFSET              ;PREPARE FOR LOAD
217 000646  103002                          BCC     100$                ;OK, CONTINUE
218 000650  000167  000352                  JMP     PR2X
219                             ;
220 000654  005046              100$:       CLR     -(SP)               ;START MICROCODE AT 0
221 000656                                  CALL    SEQCS
222                             ;
223 000662  005046                          CLR     -(SP)               ;REINHIBIT BRANCH CONTROL REG
224 000664                                  CALL    CPCR
225 000670  005046                          CLR     -(SP)
226 000672                                  CALL    LBCP                ;SINGLE CLOCK TO REINHIBIT BRANCH REGISTER
227 000676  012746  000377                  MOV     #377,-(SP)          ;SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
228 000702                                  CALL    SEQMM
```

```
                                        ;
000706  005046                          CLR     -(SP)                   ;REINHIBIT BRANCH CONTROL REG
000710                                  CALL    MRPCR
000714  005046                          CLR     -(SP)
000716                                  CALL    LBMRP                   ;SINGLE CLOCK TO REINHIBIT BRANCH REGISTER
000722  012767  001000  176422          MOV     #Q$REBK,QR$CR2          ;RE-ARM INTERRUPTS
000730  012767  120000  176422          MOV     #<Q$SM+Q$ENOP>,QR$CR2   ;SET SEARCH MODE + ENABLE INTERRUPTS
000736  012746  000360                  MOV     #Q$CSEL,-(SP)           ;CLEAR ALL SELECTIONS
000742  052716  001001                  BIS     #<Q$LBD+Q$LBP>,(SP)     ;CLEAR DRIVE AND PULSE
000746  052716  030000                  BIS     #Q$CNC,(SP)             ;CLEAR CP NO-CLOCK
000752  005046                          CLR     -(SP)                   ;SET NOTHING
000754                                  CALL    CSR1
                                        ;
000760  012767  000005  176424  PRCD:   MOV     #Q$RDCD,QR$LBR          ;MOVE ATTN CODE TO LOD BUS REG
000766  012767  120100  176422          MOV     #<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2 ;SET ATTN CODE READY
000774  016701  176422          1$:     MOV     QR$CR2,R1               ;READ CSR2
001000  032701  000100                  BIT     #Q$ATTN,R1              ;ATTN CLEAR
001004  001373                          BNE     1$                      ;NO, READ AGAIN
                                        ;
001006  016767  000000G 176424          MOV     MSTR2,QR$LBR            ;CD MEMORY START ADDRESS
001014  012767  120040  176422          MOV     #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
001022  016701  176422          2$:     MOV     QR$CR2,R1               ;READ CSR2
001026  032701  000040                  BIT     #Q$CCCP,R1              ;IS CC TO CP CLEAR
001032  001373                          BNE     2$                      ;NO, READ AGAIN
                                        ;
001034  012767  000001  176424          MOV     #1,QR$LBR               ;TRANSFER COUNT = 1 WORD
001042  012767  120040  176422          MOV     #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
001050  016701  176422          3$:     MOV     QR$CR2,R1               ;READ CSR2
001054  032701  000040                  BIT     #Q$CCCP,R1              ;IS CC TO CP CLEAR
001060  001373                          BNE     3$                      ;NO, READ AGAIN
                                        ;
001062  012767  000000G 176424          MOV     #DATA1,QR$LBR           ;CC MEMORY DATA BUFFER
001070  012767  120040  176422          MOV     #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
                                        ;
                                        ;       WAIT FOR INTERRUPT FROM CP
                                        ;
001076                                  WTSE$S  #EFN.3
                                        ;
001110                                  CLEF$S  #EFN.3
                                        ;
                                        ;       RE-ARM INTERRUPTS
                                        ;
001122  012767  100400  176422          MOV     #<Q$SM+Q$CHB>,QR$CR2    ;CLEAR INTERRUPT (USE HIT BUFFER INT)
001130  012767  101000  176422          MOV     #<Q$SM+Q$REBK>,QR$CR2   ;RE-ARM
001136  012767  160000  176422          MOV     #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2 ;ENABLE
                                        ;
001144                                  CALL    PRDATA                  ;PRINT MEMORY CONTENTS
001150  103417                          BCS     PRCDX                   ;END OF MEMORY, EXIT
                                        ;
001152  066767  000000G 000000G         ADD     INCVAL,MSTR2            ;ADVANCE ADDRESS
001160  026767  000000G 000000G         CMP     MSTR2,MEND              ;HAS UPPER MEMORY LIMIT BEEN REACHED
001166  101674                          BLOS    PRCD                    ;NO, CONTINUE
001170  032767  000000G 000000G         BIT     #LOOP,BASE              ;LOOP ON?
001176  001404                          BEQ     PRCDX                   ;NO
001200  016767  000000G 000000G         MOV     MSTRT,MSTR2             ;INIT START ADDRESS
001206  000664                          BR      PRCD                    ;AND REPEAT PRINT
                                        ;
```

```
 286 001210  005046                    PRCDX:  CLR     -(SP)                   ;CLEAR NOTHING IN CSR1
 287 001212  012746  176000                     MOV     #0$NCLK,-(SP)           ;SET NO-CLOCKS
 288 001216                                     CALL    CSR1
 289 001222  005067  176422                     CLR     QR$CR2                  ;SET LOAD MODE
 290                                    ;
 291 001226                             PR2X:
 292 001226  042767  000000C 000000G            BIC     #<ONCE+OUT>,BASE        ;CLEAR PRINT CONTROL FLAGS
 293 001234                                     CALL    KILL                    ;KILL AST
 294 001240  000167  000000G                    JMP     CPXX
```

```
                                        ;
                                        ;
                                        ;       PRINT BUFFER COMMAND
                                        ;
                                        ;
001244                                  PB2::
001244                                          CALL    FIND                    ;LOCATE NON-BLANK IN COMMAND LINE
001250  103004                                 BCC     1$                      ;OK, CONTINUE
001252                                          CALL    ERR4                    ;'MISSING OPERAND'
001256  000167  000272                         JMP     PB2X                    ;EXIT
                                        ;
001262  012700  000003                  1$:     MOV     #PB2LN,R0               ;NUMBER OF TABLE ENTRIES
001266  012702  000010'                         MOV     #PB2TBL,R2              ;POINT TO TABLE
001272                                          CALL    SCAN                    ;MATCH AGAINST COMMAND LINE
001276  103004                                 BCC     2$                      ;MATCH WAS MADE
001300                                          CALL    ERR7                    ;'INVALID MEMORY MNEMONIC'
001304  000167  000244                         JMP     PB2X
                                        ;
001310  000171  000000                  2$:     JMP     @(R1)                   ;JUMP TO ROUTINE
                                        ;
                                        ;
                                        ;       PRINT HIT BUFFER
                                        ;
                                        ;
001314                                  PB2HB::
001314                                          CALL    HANG                    ;HOW TO STOP PRINT
001320  012703  000000G                         MOV     #HLB,R3                 ;R3 -> HIT LIST BUFFER
001324  012700  000200                         MOV     #128.,R0                ;NUMBER OF WORDS IN BUFFER
001330  012705  000000G                 1$:     MOV     #PRINT,R5               ;R5 -> PRINT LINE
001334  012702  000014                         MOV     #12.,R2                 ;NUMBER OF WORDS PER PRINT LINE
001340  012301                          2$:     MOV     (R3)+,R1                ;LOAD WORD TO PRINT INTO R1
001342                                          CALL    UNPK                    ;CONVERT TO ASCII HEX
001346  005300                                 DEC     R0                      ;FINISHED WITH TABLE ?
001350  001413                                 BEQ     4$                      ;YES, EXIT
001352  005302                                 DEC     R2                      ;FINISHED WITH CURRENT PRINT LINE ?
001354  001402                                 BEQ     3$                      ;YES, WRITE TO CONSOLE
001356  005205                                 INC     R5                      ;BUMP PRINT LINE POINTER
001360  000767                                 BR      2$                      ;GET NEXT WORD
                                        ;
001362                                  3$:     CALL    CONSOL                  ;WRITE TO TERMINAL
001366  032767  000000G 000000G                BIT     #LOOP,BASE              ;CONTINUE WRITING
001374  001467                                 BEQ     PB2X                    ;NO, EXIT
001376  000754                                 BR      1$                      ;INITIALIZE NEXT PRINT LINE
001400                                  4$:     CALL    CONSOL                  ;WRITE TO TERMINAL
001404  000463                                 BR      PB2X
                                        ;
                                        ;
                                        ;       PRINT OUT HRL BUFFER
                                        ;
                                        ;
001406                                  PB2HL::
001406                                          CALL    HANG                    ;HOW TO STOP PRINT
001412  012703  000000G                         MOV     #HRL,R3                 ;R3 -> HRL BUFFER
001416  012700  000200                         MOV     #128.,R0                ;NUMBER OF WORDS IN BUFFER
001422  012705  000000G                 1$:     MOV     #PRINT,R5               ;R5 -> PRINT LINE
001426  012702  000014                         MOV     #12.,R2                 ;NUMBER OF WORDS PER PRINT LINE
001432  012301                          2$:     MOV     (R3)+,R1                ;LOAD WORD TO PRINT INTO R1
```

```
353 001434                                      CALL    UNPK                    ;CONVERT TO ASCII HEX
354 001440  005300                              DEC     R0                      ;FINISHED WITH TABLE ?
355 001442  001413                              BEQ     4$                      ;YES, EXIT
356 001444  005302                              DEC     R2                      ;FINISHED WITH CURRENT PRINT LINE ?
357 001446  001402                              BEQ     3$                      ;YES, WRITE TO CONSOLE
358 001450  005205                              INC     R5                      ;BUMP PRINT LINE POINTER
359 001452  000767                              BR      2$                      ;GET NEXT WORD
360                                     ;
361 001454                              3$:     CALL    CONSOL                  ;WRITE TO TERMINAL
362 001460  032767  000000G 000000G            BIT     #LOOP,BASE              ;CONTINUE WRITING
363 001466  001432                              BEQ     PB2X                    ;NO, EXIT
364 001470  000754                              BR      1$                      ;INITIALIZE NEXT PRINT LINE
365 001472                              4$:     CALL    CONSOL                  ;WRITE TO TERMINAL
366 001476  000426                              BR      PB2X
367                                     ;
368                                     ;
369                                     ;       PRINT OUT BCL
370                                     ;
371                                     ;
372 001500                              PB2BL::
373 001500  012703  000000G                     MOV     #BCL,R3                 ;R3 -> BCL BUFFER
374 001504  012700  000015                      MOV     #13.,R0                 ;NUMBER OF WORDS IN BUFFER
375 001510  012705  000000G             1$:     MOV     #PRINT,R5               ;R5 -> PRINT LINE
376 001514  012702  000007                      MOV     #7.,R2                  ;NUMBER OF WORDS PER PRINT LINE
377 001520  012301                      2$:     MOV     (R3)+,R1                ;LOAD WORD TO PRINT INTO R1
378 001522                                      CALL    UNPK                    ;CONVERT TO ASCII HEX
379 001526  005300                              DEC     R0                      ;FINISHED WITH TABLE ?
380 001530  001407                              BEQ     4$                      ;YES, EXIT
381 001532  005302                              DEC     R2                      ;FINISHED WITH CURRENT PRINT LINE ?
382 001534  001402                              BEQ     3$                      ;YES, WRITE TO CONSOLE
383 001536  005205                              INC     R5                      ;BUMP PRINT LINE POINTER
384 001540  000767                              BR      2$                      ;GET NEXT WORD
385                                     ;
386 001542                              3$:     CALL    CONSOL                  ;WRITE TO TERMINAL
387 001546  000760                              BR      1$                      ;INITIALIZE NEXT PRINT LINE
388 001550                              4$:     CALL    CONSOL                  ;WRITE TO TERMINAL
389                                     ;
390 001554                              PB2X:
391 001554                                      CALL    KILL                    ;KILL AST
392 001560  000167  000000G                     JMP     CPXX
393                                     ;
394         000001                              .END
```

```
ALUCKE= 040000
ALUOE = 004000
A01   = 010000
BASE  = ****** GX
BCL   = ****** GX
BINWD = ****** GX
BITVAL= 000000
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BUFSET= ****** GX
BUFS4 = ****** GX
BYTE0 = 000000
BYTE1 = 000001
BYTE10= 000012
BYTE11= 000013
BYTE12= 000014
BYTE13= 000015
BYTE14= 000016
BYTE15= 000017
BYTE16= 000020
BYTE17= 000021
BYTE18= 000022
BYTE19= 000023
BYTE2 = 000002
BYTE20= 000024
BYTE21= 000025
BYTE22= 000026
BYTE23= 000027
BYTE24= 000030
BYTE25= 000031
BYTE26= 000032
BYTE27= 000033
BYTE28= 000034
BYTE29= 000035
BYTE3 = 000003
BYTE30= 000036
BYTE31= 000037
BYTE32= 000040
BYTE33= 000041
BYTE34= 000042
BYTE35= 000043
BYTE36= 000044
BYTE37= 000045
BYTE38= 000046
BYTE39= 000047
BYTE4 = 000004
BYTE40= 000050
BYTE41= 000051
BYTE42= 000052
BYTE43= 000053
BYTE44= 000054
BYTE45= 000055
BYTE46= 000056
BYTE47= 000057
BYTE48= 000060
BYTE49= 000061
BYTE5 = 000005
BYTE50= 000062
BYTE51= 000063
BYTE52= 000064
BYTE53= 000065
BYTE54= 000066
BYTE55= 000067
BYTE56= 000070
BYTE57= 000071
BYTE58= 000072
BYTE59= 000073
BYTE6 = 000006
BYTE60= 000074
BYTE61= 000075
BYTE62= 000076
BYTE63= 000077
BYTE64= 000100
BYTE65= 000101
BYTE66= 000102
BYTE67= 000103
BYTE68= 000104
BYTE69= 000105
BYTE7 = 000007
BYTE70= 000106
BYTE71= 000107
BYTE72= 000110
BYTE73= 000111
BYTE74= 000112
BYTE75= 000113
BYTE76= 000114
BYTE77= 000115
BYTE78= 000116
BYTE79= 000117
BYTE8 = 000010
BYTE80= 000120
BYTE81= 000121
BYTE82= 000122
BYTE83= 000123
BYTE84= 000124
BYTE85= 000125
BYTE86= 000126
BYTE87= 000127
BYTE88= 000130
BYTE89= 000131
BYTE9 = 000011
BYTE90= 000132
BYTE91= 000133
BYTE92= 000134
BYTE93= 000135
BYTE94= 000136
BYTE95= 000137
BYTE96= 000140
BYTE97= 000141
BYTE98= 000142
BYTE99= 000143
BYTVAL= 000144
CBKALL= 001000
CBKCLK= 000400
CDHIGH= ****** GX
CDLOW = ****** GX
CNOBRE= 100000
CONSOL= ****** GX
CPCCEN= 010000
CPCR  = ****** GX
CPLB  = ****** GX
CPREAD= 040000
CPWRTE= 020000
CPXX  = ****** GX
CSADRD= 000004
CSEOCI= 100000
CSHIGH= ****** GX
CSLOW = ****** GX
CSOE  = 000040
CSR1  = ****** GX
CSWRTE= 000100
DATA1 = ****** GX
DATA2 = ****** GX
DATA3 = ****** GX
DATA4 = ****** GX
DBR.RD= 000001
DB$CPP= 001457
DB$SPT= 000026
DB$TPC= 000023
DISPGS= 100000
DMAAWR= 000005
DMARRD= 000003
DMARWR= 000004
EFN.3 = ****** GX
ENBR  = 010000
ERR11 = ****** GX
ERR4  = ****** GX
ERR5  = ****** GX
ERR7  = ****** GX
FIND  = ****** GX
HANG  = ****** GX
HLB   = ****** GX
HR_   = ****** GX
INCVAL= ****** GX
KILL  = ****** GX
LBCP  = ****** GX
LBMRP = ****** GX
LOC.EN= 000100
LOC.WA= 040000
LOC.WB= 100000
LOOP  = ****** GX
MAREN1= 000001
MAREN2= 004000
MARLOD= 010000
MAROUT= 000002
MAR.LO= 002000
MAR.OU= 000040
MBKALL= 001000
MBKCLK= 000400
MEND  = ****** GX
MMADRD= 000100
MMLEFT= 000002
MMOE  = 000004
MMWRTE= 000010
MNOBRE= 100000
MREN1 = 000001
MREN2 = 020000
MRPCR = ****** GX
MSTRT = ****** GX
MSTR2 = ****** GX
MSYN  = 000040
N     = 000144
ONCE  = ****** GX
OUT   = ****** GX
PACK  = ****** GX
PB2     001244RG    002
PB2BL   001500RG    002
PB2HB   001314RG    002
PB2HL   001406RG    002
PB2LN = 000003 G
PB2TBL  000010RG    002
PB2X    001554R     002
PLB   = 000010
PLC   = 000020
PLD   = 000030
PLRWR = 000200
PLR.EN= 000200
PRCD    000760R     002
PRCDX   001210R     002
PRDATA= ****** GX
PRINT = ****** GX
PR2     000024RG    002
PR2CD   000632RG    002
PR2CS   000312RG    002
PR2LN = 000002 G
PR2TBL  000000RG    002
PR2X    001226R     002
QR$CR1= 176420
QR$CR2= 176422
QR$LBR= 176424
Q$ATTN= 000100
Q$BCL = 000001
Q$CCCP= 000040
Q$CHB = 000400
Q$CHRL= 000200
Q$CLR = 000040
Q$CNC = 030000
Q$CP  = 000060
Q$CPCC= 000010
Q$CP2 = 000260
Q$CSC = 010000
Q$CSEL= 000360
Q$CSET= 000002
Q$CSP = 020000
Q$DMA = 000001
Q$ENBK= 040000
Q$ENOP= 020000
Q$FAL = 004000
Q$FC  = 000045
Q$FO  = 000044
Q$FP  = 000046
Q$HBF = 000002
Q$ICP = 000006
Q$IHB = 000003
Q$IHRL= 000002
Q$IMRP= 000007
Q$LBD = 001000
Q$LBDP= 001001
Q$LBP = 000001
Q$LDCD= 000003
Q$LDMD= 000004
Q$LDPP= 002000
Q$LHP = 010000
Q$MNC = 140000
Q$MR  = 000052
Q$MRP = 000040
Q$MRP2= 000240
Q$MSC = 040000
Q$MSET= 000004
Q$MSP = 100000
Q$NCLK= 176000
Q$PP  = 000100
Q$PPSW= 000320
Q$PP2 = 000300
Q$QHLT= 000013
Q$QL  = 000043
Q$QLA = 000053
Q$QLB = 000054
Q$QLR = 000001
Q$QW  = 000042
Q$RDCD= 000005
Q$RDMD= 000006
Q$REBK= 001000
Q$RNC = 006000
Q$RSC = 004000
Q$RSET= 000010
Q$SM  = 100000
Q$SP  = 000120
Q$SP2 = 000340
RGQ.EN= 000200
RGQ.VA= 020000
```

```
RTNPT = ****** GX      T$DISK= 000200      WORD1  = 000002      WORD40= 000120      WORD71= 000216
SCAN  = ****** GX      T$DRD = 000004      WORD10= 000024      WORD41= 000122      WORD72= 000220
SEQCS = ****** GX      T$EMEM= 010000      WORD11= 000026      WORD42= 000124      WORD73= 000222
SEQMM = ****** GX      T$FSAA= 000000      WORD12= 000030      WORD43= 000126      WORD74= 000224
SEQ.CI= 000010         T$FSAB= 000004      WORD13= 000032      WORD44= 000130      WORD75= 000226
S$CLR = 000000         T$FSAC= 000014      WORD14= 000034      WORD45= 000132      WORD76= 000230
S$LA  = 000001         T$FSB2= 000010      WORD15= 000036      WORD46= 000134      WORD77= 000232
S$QB  = 000005         T$IB  = 000026      WORD16= 000040      WORD47= 000136      WORD78= 000234
S$QR  = 000006         T$IBAR= 000024      WORD17= 000042      WORD48= 000140      WORD79= 000236
S$QX  = 000004         T$IBE = 020000      WORD18= 000044      WORD49= 000142      WORD8  = 000020
S$SR  = 000007         T$IBF = 040000      WORD19= 000046      WORD5  = 000012      WORD80= 000240
S$S1  = 000010         T$ICD = 000040      WORD2  = 000004     WORD50= 000144      WORD81= 000242
S$S2  = 000014         T$MODE= 004000      WORD20= 000050      WORD51= 000146      WORD82= 000244
TD$CTR= 176370         T$OB  = 000036      WORD21= 000052      WORD52= 000150      WORD83= 000246
TD$CTW= 176360         T$OBE = 004000      WORD22= 000054      WORD53= 000152      WORD84= 000250
TD$INL= 004000         T$OBF = 010000      WORD23= 000056      WORD54= 000154      WORD85= 000252
TD$MEM= 000270         T$OBRA= 000034      WORD24= 000060      WORD55= 000156      WORD86= 000254
TD$OAR= 176344         T$OBWA= 000032      WORD25= 000062      WORD56= 000160      WORD87= 000256
TD$OTR= 176346         T$OUTA= 100000      WORD26= 000064      WORD57= 000162      WORD88= 000260
TD$ORD= 000274         T$RBDO= 000200      WORD27= 000066      WORD58= 000164      WORD89= 000262
TD$SW = 176376         T$RNB = 000040      WORD28= 000070      WORD59= 000166      WORD9  = 000022
TD$TAR= 176372         T$RSET= 040000      WORD29= 000072      WORD6  = 000014     WORD90= 000264
TD$TAW= 176362         T$SC  = 000022      WORD3  = 000006     WORD60= 000170      WORD91= 000266
TD$TDR= 176374         T$SCLK= 020000      WORD30= 000074      WORD61= 000172      WORD92= 000270
TD$TDW= 176364         T$SEG1= 000000      WORD31= 000076      WORD62= 000174      WORD93= 000272
T$AD  = 000020         T$SEG2= 000001      WORD32= 000100      WORD63= 000176      WORD94= 000274
T$BA  = 000002         T$SEG3= 000002      WORD33= 000102      WORD64= 000200      WORD95= 000276
T$BD  = 000010         T$SO  = 000001      WORD34= 000104      WORD65= 000202      WORD96= 000300
T$BSO = 100000         T$UBUS= 100000      WORD35= 000106      WORD66= 000204      WORD97= 000302
T$BT  = 000020         T$1CLK= 000400      WORD36= 000110      WORD67= 000206      WORD98= 000304
T$BTAR= 000030         T$8BEN= 000020      WORD37= 000112      WORD68= 000210      WORD99= 000306
T$BTD = 002000         UBD.IN= 000020      WORD38= 000114      WORD69= 000212      WRDVAL= 000310
T$CD  = 000100         UNPK  = ****** GX   WORD39= 000116      WORD7  = 000016     XTREAD= 001000
T$CLK = 002000         WORD0  = 000000     WORD4  = 000010     WORD70= 000214      XTWRTE= 000400

. ABS.  000000    000
        000000    001
CPPR    001564    002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3671 WORDS  ( 15 PAGES)
DYNAMIC MEMORY:  4916 WORDS  ( 18 PAGES)
ELAPSED TIME:  00:00:50
CPPR,CPPR/-SP=[20,1]IM,[20,1]CPPR
```

```
                                        .TITLE  BCE
  000000                                .PSECT  BCE
                                        .LIST   MEB
                                ;
                                ;
                                ;       HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
                                ;       BUS CONTROL ELEMENT TEST ROUTINES
                                ;
                                ;       PARSE THE COMMAND LINE AND TRANSFER CONTROL TO ONE OF THE
                                ;       BCE SUB-MODULES OR BACK TO QMAIN. CONTROL COULD HAVE BEEN
                                ;       PASSED TO BCE IN ONE OF THREE WAYS:
                                ;       1. FROM QMAIN IF BCE WAS SELECTED AS THE FIRST PROCESSOR
                                ;          UPON ENTRY TO THE PROGRAM
                                ;          >BC ST 0 C1
                                ;       2. FROM QMAIN IF BCE WAS SELECTED FROM ONE OF QMAIN'S
                                ;          OTHER SUB-MODULES
                                ;          CP>BC ST 0 C1
                                ;       3. ON RETURN FROM ONE OF BCE'S SUB-MODULES
                                ;          BC>ST 0 C1
                                ;
                                ;       BCE PARSES AT THE SECOND LEVEL OF CONTROL (SEE NOTES AT
                                ;       QMAIN ON LEVELS OF CONTROL). IN THE EXAMPLES ABOVE, BCE
                                ;       WOUST PARSE 'LD' AND TRANSFER CONTROL TO THE BCE SUB-
                                ;       MODULE BCREST. IF BCE SHOULD ENCOUNTER A STRING WHICH IS
                                ;       NOT A VALID COMMAND MNEMONIC, BCE CONSIDERS THE STRING TO
                                ;       BE A PROCESSOR MNEMONIC AND RETURNS CONTROL TO QMAIN. EG:
                                ;       BC>CP LD CS 0
                                ;       THE STRING 'CP' IS NOT A VALID BCE (SECOND LEVEL) COMMAND.
                                ;       BCE RETURNS CONTROL TO QMAIN WHICH IN TURN WILL TRANSFER
                                ;       CONTROL TO ITS SUB-MODULE CP.
                                ;
                                ;       BCE SUB-MODULES:
                                ;       BCREST  ALL BCE COMMANDS
                                ;
                                ;
  000000                        BCTBL:
  000000     123      124               .ASCII  /ST/            ;STORE INTO A REG
  000002  000000G                       .WORD   ST3
  000004     122      105               .ASCII  /RE/            ;READ FROM A REG
  000006  000000G                       .WORD   RE3
  000010     122      123               .ASCII  /RS/            ;RESET HQR
  000012  000000G                       .WORD   RS3
          000003                BCNUM   =       <.-BCTBL>/4
```

```
                                        ;
                                        ;
                                        ;       BCE ROUTINES
                                        ;
                                        ;
50 000014                               BCE::
51 000014  004767  000000G              JSR     PC,FIND         ;LOCATE THE COMMAND IN THE COMMAND LINE
52 000020  103003                       BCC     1$              ;OK, CONTINUE
53 000022  004767  000000G              JSR     PC,ERR3
54 000026  000422                       BR      BCEXX
55 000030  022700  000002       1$:     CMP     #2,R0           ;COMMANDS ARE 2 CHARS
56 000034  001403                       BEQ     2$
57 000036  004767  000000G              JSR     PC,ERR8
58 000042  000414                       BR      BCEXX           ;TRY AGAIN
59                                      ;
60 000044  012700  000003       2$:     MOV     #BCNUM,R0       ;R0 = NUMBER OF COMMANDS
61 000050  012702  000000'              MOV     #BCTBL,R2       ;R2 -> TABLE OF COMMAND MNEMONICS
62 000054  004767  000000G              JSR     PC,SCAN         ;FIND MATCH IN TABLE
63 000060  103003                       BCC     3$              ;OK, CONTINUE
64 000062  004767  000000G              JSR     PC,ERR12        ;COMMAND NOT IN TABLE
65 000066  000402                       BR      BCEXX           ;TRY AGAIN
66                                      ;
67                                      ;       JUMP TO THE ROUTINE THAT GOVERNS THE COMMAND
68                                      ;
69 000070  000171  000000       3$:     JMP     @(R1)
70                                      ;
71                                      ;
72                                      ;       LOCAL BCE LOOP, INCLUDE PROMPT FOR BCE COMMAND,
73                                      ;
74                                      ;
75 000074                               BCEXX::
76 000074  012767  041502  000000C      MOV     #"BC,GCMBLK+G.DPRM+2 ;MOVE BCE NAME TO GCM;BLOCK
77 000102  004767  000000G              JSR     PC,GCONLY       ;PROMPT
78 000106  004767  000000G              JSR     PC,FIND         ;LOCATE THE COMMAND IN THE COMMAND LINE
79 000112  103003                       BCC     1$              ;OK, CONTINUE
80 000114  004767  000000G              JSR     PC,ERR3
81 000120  000765                       BR      BCEXX
82 000122  022700  000002       1$:     CMP     #2,R0           ;COMMANDS ARE 2 CHARS
83 000126  001403                       BEQ     2$
84 000130  004767  000000G              JSR     PC,ERR8
85 000134  000757                       BR      BCEXX           ;TRY AGAIN
86                                      ;
87 000136  012700  000003       2$:     MOV     #BCNUM,R0       ;R0 = NUMBER OF COMMANDS
88 000142  012702  000000'              MOV     #BCTBL,R2       ;R2 -> TABLE OF COMMAND MNEMONICS
89 000146  004767  000000G              JSR     PC,SCAN         ;FIND MATCH IN TABLE
90 000152  103002                       BCC     3$              ;OK, CONTINUE
91 000154  000167  000000G              JMP     COMXX           ;RETURN TO 'MAIN', LOOK FOR PROCESSOR MNEMONIC
92                                      ;
93                                      ;       JUMP TO THE ROUTINE THAT GOVERNS THE COMMAND
94                                      ;
95 000160  000171  000000       3$:     JMP     @(R1)
96                                      ;
97         000001                       .END
```

```
ALUCKE= 040000
ALUOE = 004000
A01    = 010000
BCE     000014RG    002
BCEXX   000074RG    002
BCNUM = 000003
BCTBL   000000R     002
BITVAL= 000000
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BYTE0 = 000000
BYTE1 = 000001
BYTE10= 000012
BYTE11= 000013
BYTE12= 000014
BYTE13= 000015
BYTE14= 000016
BYTE15= 000017
BYTE16= 000020
BYTE17= 000021
BYTE18= 000022
BYTE19= 000023
BYTE2 = 000002
BYTE20= 000024
BYTE21= 000025
BYTE22= 000026
BYTE23= 000027
BYTE24= 000030
BYTE25= 000031
BYTE26= 000032
BYTE27= 000033
BYTE28= 000034
BYTE29= 000035
BYTE3 = 000003
BYTE30= 000036
BYTE31= 000037
BYTE32= 000040
BYTE33= 000041
BYTE34= 000042
BYTE35= 000043
BYTE36= 000044
BYTE37= 000045
BYTE38= 000046

BYTE39= 000047
BYTE4 = 000004
BYTE40= 000050
BYTE41= 000051
BYTE42= 000052
BYTE43= 000053
BYTE44= 000054
BYTE45= 000055
BYTE46= 000056
BYTE47= 000057
BYTE48= 000060
BYTE49= 000061
BYTE5 = 000005
BYTE50= 000062
BYTE51= 000063
BYTE52= 000064
BYTE53= 000065
BYTE54= 000066
BYTE55= 000067
BYTE56= 000070
BYTE57= 000071
BYTE58= 000072
BYTE59= 000073
BYTE6 = 000006
BYTE60= 000074
BYTE61= 000075
BYTE62= 000076
BYTE63= 000077
BYTE64= 000100
BYTE65= 000101
BYTE66= 000102
BYTE67= 000103
BYTE68= 000104
BYTE69= 000105
BYTE7 = 000007
BYTE70= 000106
BYTE71= 000107
BYTE72= 000110
BYTE73= 000111
BYTE74= 000112
BYTE75= 000113
BYTE76= 000114
BYTE77= 000115
BYTE78= 000116
BYTE79= 000117
BYTE8 = 000010
BYTE80= 000120
BYTE81= 000121
BYTE82= 000122
BYTE83= 000123
BYTE84= 000124
BYTE85= 000125
BYTE86= 000126
BYTE87= 000127
BYTE88= 000130
BYTE89= 000131
BYTE9 = 000011

BYTE90= 000132
BYTE91= 000133
BYTE92= 000134
BYTE93= 000135
BYTE94= 000136
BYTE95= 000137
BYTE96= 000140
BYTE97= 000141
BYTE98= 000142
BYTE99= 000143
BYTVAL= 000144
CBKALL= 001000
CBKCLK= 000400
CNOBRE= 100000
COMXX = ****** GX
CPCCEN= 010000
CPREAD= 040000
CPWRTE= 020000
CSADRD= 000004
CSEQCI= 100000
CSOE  = 000040
CSWRTE= 000100
DBR.RD= 000001
DB$CPP= 001457
DB$SPT= 000026
DB$TPC= 000023
DISPGS= 100000
DMAAWR= 000005
DMARRD= 000003
DMARWR= 000004
ENBR  = 010000
ERR12 = ****** GX
ERR3  = ****** GX
ERR8  = ****** GX
FIND  = ****** GX
GCMBLK= ****** GX
GCONLY= ****** GX
G.DPRM= ****** GX
LOC.EN= 000100
LOC.WA= 040000
LOC.WB= 100000
MAREN1= 000001
MAREN2= 004000
MARLOD= 010000
MAROUT= 000002
MAR.LO= 002000
MAR.OU= 000040
MBKALL= 001000
MBKCLK= 000400
MMADRD= 000100
MMLEFT= 000002
MMOE  = 000004
MMWRTE= 000010
MNOBRE= 100000
MREN1 = 000001
MREN2 = 020000
MSYN  = 000040

N     = 000144
PLB   = 000010
PLC   = 000020
PLD   = 000030
PLRWR = 000200
PLR.EN= 000200
QR$CR1= 176420
QR$CR2= 176422
QR$LBR= 176424
Q$ATTN= 000100
Q$BCL = 000001
Q$CCCP= 000040
Q$CHB = 000400
Q$CHRL= 000200
Q$CLR = 000040
Q$CNC = 030000
Q$CP  = 000060
Q$CPCC= 000010
Q$CP2 = 000260
Q$CSC = 010000
Q$CSEL= 000360
Q$CSET= 000002
Q$CSP = 020000
Q$DMA = 000001
Q$ENBK= 040000
Q$ENOP= 020000
Q$FAL = 004000
Q$FC  = 000045
Q$FO  = 000044
Q$FP  = 000046
Q$HBF = 000002
Q$ICP = 000006
Q$IHB = 000003
Q$IHRL= 000002
Q$IMRP= 000007
Q$LBD = 001000
Q$LBDP= 001001
Q$LBP = 000001
Q$LDCD= 000003
Q$LDMD= 000004
Q$LDPP= 002000
Q$LHP = 010000
Q$MNC = 140000
Q$MR  = 000052
Q$MRP = 000040
Q$MRP2= 000240
Q$MSC = 040000
Q$MSET= 000004
Q$MSP = 100000
Q$NCLK= 176000
Q$PP  = 000100
Q$PPSW= 000320
Q$PP2 = 000300
Q$QHLT= 000013
Q$QL  = 000043
Q$QLA = 000053
Q$QLB = 000054

Q$QLR = 000001
Q$QW  = 000042
Q$RDCD= 000005
Q$RDMD= 000006
Q$REBK= 001000
Q$RNC = 006000
Q$RSC = 004000
Q$RSET= 000010
Q$SM  = 100000
Q$SP  = 000120
Q$SP2 = 000340
RE3   = ****** GX
RGQ.EN= 000200
RGQ.VA= 020000
RS3   = ****** GX
SCAN  = ****** GX
SEQ.CI= 000010
ST3   = ****** GX
S$CLR = 000000
S$LA  = 000001
S$QB  = 000005
S$QR  = 000006
S$QX  = 000004
S$SR  = 000007
S$S1  = 000010
S$S2  = 000014
TD$CTR= 176370
TD$CTW= 176360
TD$INL= 004000
TD$MEM= 000270
TD$OAR= 176344
TD$OTR= 176346
TD$QRD= 000274
TD$SW = 176376
TD$TAR= 176372
TD$TAW= 176362
TD$TDR= 176374
TD$TDW= 176364
T$AD  = 000020
T$BA  = 000002
T$BD  = 000010
T$BSO = 100000
T$BT  = 000020
T$BTAR= 000030
T$BTD = 002000
T$CD  = 000100
T$CLK = 002000
T$DISK= 000200
T$DRD = 000004
T$EMEM= 010000
T$FSAA= 000000
T$FSAB= 000004
T$FSAC= 000014
T$FSB2= 000010
T$IB  = 000026
T$IBAR= 000024
T$IBE = 020000
```

SYMBOL·TABLE·

```
T$IBF· =·040000          WORD11=·000026          WORD34=·000104          WORD57=·000162·         WORD8 =·000020
T$ICD· =·000040          WORD12=·000030          WORD35=·000106          WORD58=·000164          WORD80=·000240
T$MODE=·004000           WORD13=·000032·         WORD36=·000110          WORD59=·000166          WORD81=·000242
T$OB·  =·000036          WORD14=·000034          WORD37=·000112·         WORD6 =·000014          WORD82=·000244·
T$OBE· =·004000          WORD15=·000036          WORD38=·000114          WORD60=·000170          WORD83=·000246
T$OBF· =·010000          WORD16=·000040          WORD39=·000116          WORD61=·000172·         WORD84=·000250
T$OBRA=·000034           WORD17=·000042·         WORD4 =·000010          WORD62=·000174          WORD85=·000252
T$OBWA=·000032·          WORD18=·000044          WORD40=·000120          WORD63=·000176          WORD86=·000254
T$OUTA=·100000           WORD19=·000046          WORD41=·000122·         WORD64=·000200          WORD87=·000256
T$RBDO=·000200           WORD2·=·000004          WORD42=·000124          WORD65=·000202·         WORD88=·000260
T$RNB· =·000040          WORD20=·000050          WORD43=·000126          WORD66=·000204          WORD89=·000262
T$RSET=·040000           WORD21=·000052·         WORD44=·000130          WORD67=·000206          WORD9 =·000022·
T$SC·  =·000022·         WORD22=·000054          WORD45=·000132·         WORD68=·000210          WORD90=·000264
T$SCLK=·020000           WORD23=·000056          WORD46=·000134          WORD69=·000212·         WORD91=·000266
T$SEG1=·000000           WORD24=·000060          WORD47=·000136          WORD7 =·000016          WORD92=·000270
T$SEG2=·000001           WORD25=·000062·         WORD48=·000140          WORD70=·000214          WORD93=·000272
T$SEG3=·000002·          WORD26=·000064          WORD49=·000142·         WORD71=·000216          WORD94=·000274
T$SO·  =·000001          WORD27=·000066          WORD5 =·000012·         WORD72=·000220          WORD95=·000276
T$UBUS=·100000           WORD28=·000070          WORD50=·000144          WORD73=·000222·         WORD96=·000300
T$1CLK=·000400           WORD29=·000072·         WORD51=·000146          WORD74=·000224          WORD97=·000302
T$8BEN=·000020           WORD3 =·000006          WORD52=·000150          WORD75=·000226          WORD98=·000304
UBD.IN=·000020           WORD30=·000074          WORD53=·000152·         WORD76=·000230          WORD99=·000306
WORD0 =·000000           WORD31=·000076          WORD54=·000154          WORD77=·000232·         WRDVAL=·000310
WORD1 =·000002·          WORD32=·000100          WORD55=·000156          WORD78=·000234          XTREAD=·001000
WORD10=·000024           WORD33=·000102·         WORD56=·000160          WORD79=·000236          XTWRTE=·000400

. ABS.  000000     000
        000000     001
BCE·    000164     002·
ERRORS·DETECTED:  0

VIRTUAL·MEMORY·USED:  3103 WORDS·  (·13 PAGES)
DYNAMIC·MEMORY:  3860 WORDS·  (  14 PAGES)
ELAPSED·TIME:  00:00:42·
BCE,BCE/-SP=[20,1]IM,[20,1]BCE
```

```
 1                                              .TITLE  BCREST
 2 000000                                       .PSECT  BCREST
 3                                              .LIST   MEB
 4                                      ;
 5                                      ;
 6                                      ;       HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
 7                                      ;       BUS CONTROL ELEMENT TEST ROUTINES
 8                                      ;
 9                                      ;       COMMANDS:
10                                      ;       ST      STORE INTO A REGISTER
11                                      ;       RE      READ A REGISTER
12                                      ;       RS      MASTER HQR RESET
13                                      ;
14                                      ;
15                                      ;       STORE
16                                      ;       TABLE OF VALID REGISTER MNEMONICS AND ASSOCIATED ROUTINE
17                                      ;       ADDRESSES
18                                      ;
19 000000                              ST3TBL::
20 000000      103     061                     .ASCII  /C1/            ;CONTROL AND STATUS REG #1
21 000002  000150'                             .WORD   ST3C1
22 000004      103     062                     .ASCII  /C2/            ;CONTROL AND STATUS REG #2
23 000006  000160'                             .WORD   ST3C2
24 000010      114     102                     .ASCII  /LB/            ;LOD BUS REG
25 000012  000170'                             .WORD   ST3LB
26             000003                  ST3LN   ==      <.-ST3TBL>/4
27                                      ;
28                                      ;       READ
29                                      ;       TABLE OF VALID REGISTER MNEMONICS AND ASSOCIATED ROUTINE
30                                      ;       ADDRESSES
31                                      ;
32 000014                              RE3TBL::
33 000014      103     061                     .ASCII  /C1/            ;CONTROL AND STATUS REG #1
34 000016  000310'                             .WORD   RE3C1
35 000020      103     062                     .ASCII  /C2/            ;CONTROL AND STATUS REG #2
36 000022  000316'                             .WORD   RE3C2
37 000024      114     102                     .ASCII  /LB/            ;LOD BUS REG
38 000026  000324'                             .WORD   RE3LB
39             000003                  RE3LN   ==      <.-RE3TBL>/4
40                                      ;
41                                      ;
42                                      ;       STORE
43                                      ;       PERFORM THIRD LEVEL PARSING
44                                      ;       EG. IN THE COMMAND:
45                                      ;       BC>ST 0 C1
46                                      ;       PARSE THE '0'
47                                      ;
48                                      ;
49 000030                              ST3::
50 000030  004767  000000G                     JSR     PC,FIND         ;FIND A NON-BLANK IN COMMAND LINE
51 000034  103004                              BCC     1$              ;OK, CONTINUE
52 000036  004767  000000G                     JSR     PC,ERR4         ;NOTHING THERE
53 000042  000167  000146                      JMP     ST3X            ;RETURN TO TOP OF LOOP (PROMPT)
54 000046  004767  000000G             1$:     JSR     PC,PACK         ;CONVERT VALUE IN COMMAND LINE TO BINARY
55 000052  103004                              BCC     2$
56 000054  004767  000000G                     JSR     PC,ERR5
57 000060  000167  000130                      JMP     ST3X
```

```
                                        ;
                                        ;       CONTINUE·PARSING·
                                        ;       FIND·THE·REGISTER·MNEMONIC·IN·THE·COMMAND·LINE·
                                        ;
000064  004767  000000G·        2$:     JSR·    PC,FIND·        ;LOCATE·A·NON-BLANK·
000070  103004                          BCC·    3$
000072· 004767  000000G·                JSR·    PC,ERR4
000076  000167  000112                  JMP·    ST3X
                                        ;
                                        ;       MATCH·THE·MNEMONIC·FROM·THE·COMMAND·LINE·AGAINST·THE·
                                        ;       TBALE·OF·VALID·MNEMONICS.
                                        ;
000102  012700  000003          3$:     MOV·    #ST3LN,R0       ;NUMBER·OF·TABLE·ENTRIES·
000106  012702· 000000'                 MOV·    #ST3TBL,R2·     ;R2·-> TABLE·
000112· 004767  000000G·                JSR·    PC,SCAN·        ;MATCH·AGAINST·COMMAND·LINE·
000116  103004                          BCC·    4$              ;OK, CONTINUE·
000120  004767  000000G·                JSR·    PC,ERR6
000124  000167  000064                  JMP·    ST3X
                                        ;
                                        ;       SAVE·THE·POINTER·TO·THE·ROUTINE·ASSOCIATED·WITH·THE·
                                        ;       REGISTER, R1 -> ROUTINE·ADDRESS.
                                        ;       CALL·ROUTINE·TO·SCAN·COMMAND·LINE·FOR·LOOP INDICATOR.
                                        ;       EG. BC>ST·0 C1 L·
                                        ;       LOOP·FLAG·WILL·BE·SET·IF·INDICATOR·IS·PRESENT.
                                        ;       JUMP·TO·ROUTINE·TO·LOAD·REGISTER.
                                        ;
000130  010167  000000G·        4$:     MOV·    R1,RTNPT·       ;SAVE·POINTER·TO·RTN·
000134  004767  000000G·                JSR·    PC,LOOPR·       ;LOOP?·
000140  016701  000000G·        ST3IN:  MOV·    RTNPT,R1        ;POINT·TO·ROUTINE·
000144  000171  000000                  JMP·    @(R1)           ;EXECUTE ROUTINE·
                                        ;
                                        ;
                                        ;       CONTROL·AND·STATUS·REG #1
                                        ;
000150                          ST3C1::
000150  016767  000000G· 176420         MOv·    BINWD,QR$CR1    ;MOVE·USER·WORD·TO CSR·#1
000156  000410                          BR      ST3LP·          ;TEST·FOR·REPEAT·
                                        ;
                                        ;       CONTROL·AND·STATUS·REG #2·
                                        ;
000160  016767  000000G· 176422· ST3C2::MOV·    BINWD,QR$CR2    ;MOVE·USER·WORD·TO CSR·#2·
000166  000404                          BR      ST3LP·          ;TEST·FOR·REPEAT·
                                        ;
                                        ;       LOD·BUS·REG·
                                        ;
000170  016767  000000G· 176424 ST3LB:: MOV·    BINWD,QR$LBR·   ;MOVE·USER·WORD·TO LOD·BUS·REG·
000176  000400                          BR      ST3LP·          ;TEST·FOR·REPEAT·
                                        ;
                                        ;
000200                          ST3LP:
000200  032767  000000G· 000000G·        BIT·    #LOOP,BASE·     ;REPEAT·?·
000206  001402·                         BEQ·    ST3X            ;NO·
000210  000167  177724                  JMP·    ST3IN·          ;AND·REPEAT·
                                        ;
000214                          ST3X:
000214  004767  000000G·                JSR·    PC,KILL·        ;KILL·AST·
000220  000167  000000G·                JMP·    BCEXX·
```

```
                                ;
                                ;
                                ;       READ
                                ;       PERFORM THIRD LEVEL PARSING
                                ;       EG. IN THE COMMAND:
                                ;       BC>RE C1
                                ;       PARSE THE 'C1'
                                ;
                                ;
000224                          RE3::
000224  004767  000000G         RE3:    JSR     PC,FIND         ;FIND A REG MNEMONIC IN COMMAND LINE
000230  103004                          BCC     1$              ;OK, CONTINUE
000232  004767  000000G                 JSR     PC,ERR4         ;NOTHING THERE
000236  000167  000136                  JMP     RE3X            ;RETURN TO TOP OF LOOP (PROMPT)
                                ;
                                ;       MATCH THE REGISTER MNEMONIC FROM THE COMMAND LINE AGAINST
                                ;       THE TABLE OF VALID ADDRESSES.
                                ;
000242  012700  000003          1$:     MOV     #RE3LN,R0       ;NUMBER OF TABLE ENTRIES
000246  012702  000014'                 MOV     #RE3TBL,R2      ;R2 -> TABLE
000252  004767  000000G                 JSR     PC,SCAN         ;MATCH AGAINST COMMAND LINE
000256  103004                          BCC     2$              ;OK, CONTINUE
000260  004767  000000G                 JSR     PC,ERR6
000264  000167  000110                  JMP     RE3X
                                ;
                                ;       SAVE THE POINTER TO THE ROUTINE ASSOCIATED WITH THE
                                ;       REGISTER, R1 -> ROUTINE ADDRESS.
                                ;       CALL ROUTINE TO SCAN COMMAND LINE FOR LOOP INDICATOR.
                                ;       EG. BC>RE C1 L
                                ;       LOOP FLAG WILL BE SET IF INDICATOR IS PRESENT
                                ;       JUMP TO ROUTINE TO LOAD REGISTER.
                                ;
000270  010167  000000G         2$:     MOV     R1,RTNPT        ;SAVE POINTER TO RTN
000274  004767  000000G                 JSR     PC,LOOPR        ;LOOP?
000300  016701  000000G         RE3IN:  MOV     RTNPT,R1        ;POINT TO ROUTINE
000304  000171  000000                  JMP     @(R1)           ;EXECUTE ROUTINE
                                ;
                                ;
                                ;       CONTROL AND STATUS REG #1
                                ;
000310                          RE3C1::
000310  016701  176420                  MOV     QR$CR1,R1       ;GET CSR #1
000314  000406                          BR      RE3PUT          ;AND PRINT
                                ;
                                ;       CONTROL AND STATUS REG #2
                                ;
000316                          RE3C2::
000316  016701  176422                  MOV     QR$CR2,R1       ;GET CSR #2
000322  000403                          BR      RE3PUT          ;AND PRINT
                                ;
                                ;       LOD BUS REG
                                ;
000324                          RE3LB::
000324  016701  176424                  MOV     QR$LBR,R1       ;GET LOD BUS REG
000330  000400                          BR      RE3PUT          ;AND PRINT
                                ;
                                ;
```

```
                                RE3PUT:...
000332                          
000332  032767  000000G 000000G        BIT     #ONCE,BASE      ;PRINTED ONCE ?
000340  001011                          BNE     1$              ;YES, SKIP
000342  052767  000000G 000000G        BIS     #ONCE,BASE      ;SET FLAG FOR PRINTED ONCE
000350  012705  000000G                MOV     #PRINT,R5       ;POINT TO PRINT LINE
000354  004767  000000G                JSR     PC,UNPK         ;CONVERT VALUE IN R1 FOR PRINTING
000360  004767  000000G                JSR     PC,CONSOL       ;PRINT ON CONSOLE
                                ;
000364  032767  000000G 000000G 1$:     BIT     #LOOP,BASE      ;REPEAT ?
000372  001402                        BEQ     RE3X            ;NO
000374  000167  177700                JMP     RE3IN           ;AND REPEAT
                                ;
000400                          RE3X:
000400  042767  000000G 000000G        BIC     #ONCE,BASE      ;CLEAR PRINT CONTROL FLAG
000406  004767  000000G                JSR     PC,KILL         ;KILL AST
000412  000167  000000G                JMP     BCEXX
                                ;
                                ;
                                ;       RESET HQR
                                ;
                                ;
000416                          RS3::
000416  005046                          CLR     -(SP)           ;CLEAR NO BITS IN CSR1  FW
000420  012746  000010                  MOV     #Q$RSET,-(SP)   ;SET RESET              FW
000424  004767  000000G                 JSR     PC,CSR1         ;CHANGE CSR1            FW
000430  012746  000010                  MOV     #Q$RSET,-(SP)   ;CLEAR RESET            FW
000434  005046                          CLR     -(SP)           ;DON'T SET ANYTHING     FW
000436  004767  000000G                 JSR     PC,CSR1         ;CHANGE CSR1            FW
000442  000167  000000G                 JMP     BCEXX
                                ;
        000001                          .END
```

SYMBOL·TABLE·


```
ALUCKE=·040000
ALUOE·=·004000
A01   =·010000
BASE·  =·****** GX·
BCEXX·=·****** GX·
BINWD·=·****** GX·
BITVAL=·000000
BIT0  =·000001
BIT1  =·000002·
BIT10 =·002000
BIT11 =·004000
BIT12·=·010000
BIT13 =·020000
BIT14 =·040000
BIT15 =·100000
BIT2·  =·000004
BIT3  =·000010
BIT4  =·000020
BIT5  =·000040
BIT6  =·000100
BIT7  =·000200
BIT8·  =·000400
BIT9  =·001000
BYTE0 =·000000
BYTE1 =·000001
BYTE10=·000012·
BYTE11=·000013
BYTE12=·000014
BYTE13=·000015
BYTE14=·000016
BYTE15=·000017
BYTE16=·000020
BYTE17=·000021
BYTE18=·000022·
BYTE19=·000023
BYTE2·=·000002·
BYTE20=·000024
BYTE21=·000025
BYTE22=·000026
BYTE23=·000027
BYTE24=·000030
BYTE25=·000031
BYTE26=·000032·
BYTE27=·000033
BYTE28=·000034
BYTE29=·000035
BYTE3 =·000003
BYTE30=·000036
BYTE31=·000037
BYTE32=·000040
BYTE33=·000041
BYTE34=·000042·
BYTE35=·000043
BYTE36=·000044
BYTE37=·000045
BYTE38=·000046
BYTE39=·000047
BYTE4 =·000004
BYTE40=·000050
BYTE41=·000051
BYTE42=·000052·
BYTE43=·000053
BYTE44=·000054
BYTE45=·000055
BYTE46=·000056
BYTE47=·000057
BYTE48=·000060
BYTE49=·000061
BYTE5 =·000005
BYTE50=·000062·
BYTE51=·000063
BYTE52=·000064
BYTE53=·000065
BYTE54=·000066
BYTE55=·000067
BYTE56=·000070
BYTE57=·000071
BYTE58=·000072·
BYTE59=·000073
BYTE6 =·000006
BYTE60=·000074
BYTE61=·000075
BYTE62=·000076
BYTE63=·000077
BYTE64=·000100
BYTE65=·000101
BYTE66=·000102·
BYTE67=·000103
BYTE68=·000104
BYTE69=·000105
BYTE7 =·000007
BYTE70=·000106
BYTE71=·000107
BYTE72=·000110
BYTE73=·000111
BYTE74=·000112·
BYTE75=·000113
BYTE76=·000114
BYTE77=·000115
BYTE78=·000116
BYTE79=·000117
BYTE8 =·000010
BYTE80=·000120
BYTE81=·000121
BYTE82=·000122·
BYTE83=·000123
BYTE84=·000124
BYTE85=·000125
BYTE86=·000126
BYTE87=·000127
BYTE88=·000130
BYTE89=·000131
BYTE9 =·000011
BYTE90=·000132·
BYTE91=·000133
BYTE92=·000134
BYTE93=·000135
BYTE94=·000136
BYTE95=·000137
BYTE96=·000140
BYTE97=·000141
BYTE98=·000142·
BYTE99=·000143
BYTVAL=·000144
CBKALL=·001000
CBKCLK=·000400
CNOBRE=·100000
CONSOL=·****** GX·
CPCCEN=·010000
CPREAD=·040000
CPWRTE=·020000
CSADRD=·000004
CSEQCI=·100000
CSOE·  =·000040
CSRi   =·****** GX·
CSWRTE=·000100
DBR.RD=·000001
DB$CPP=·001457
DB$SPT=·000026
DB$TPC=·000023
DISPGS=·100000
DMAAWR=·000005
DMARRD=·000003
DMARWR=·000004
ENBR·  =·010000
ERR4   =·****** GX·
ERR5   =·****** GX·
ERR6   =·****** GX·
FIND·  =·****** GX·
KILL·  =·****** GX·
LOC.EN=·000100
LOC.WA=·040000
LOC.WB=·100000
LOOP·  =·****** GX·
LOOPR·=·****** GX·
MAREN1=·000001
MAREN2=·004000
MARLOD=·010000
MAROUT=·000002·
MAR.LO=·002000
MAR.OU=·000040
MBKALL=·001000
MBKCLK=·000400
MMADRD=·000100
MMLEFT=·000002·
MMOE·  =·000004
MMWRTE=·000010
MNOBRE=·100000
MREN1  =·000001
MREN2·=·020000
MSYN·  =·000040
N·     =·000144
ONCE·  =·****** GX·
PACK·  =·****** GX·
PLB·   =·000010
PLC·   =·000020
PLD·   =·000030
PLRWR·=·000200
PLR.EN=·000200
PRINT·=·****** GX·
QR$CR1=·176420
QR$CR2=·176422·
QR$LBR=·176424
Q$ATTN=·000100
Q$BCL·=·000001
Q$CCCP=·000040
Q$CHB·=·000400
Q$CHRL=·000200
Q$CLR·=·000040
Q$CNC·=·030000
Q$CP·  =·000060
Q$CPCC=·000010
Q$CP2·=·000260
Q$CSC·=·010000
Q$CSEL=·000360
Q$CSET=·000002·
Q$CSP·=·020000
Q$DMA·=·000001
Q$ENBK=·040000
Q$ENOP=·020000
Q$FAL·=·004000
Q$FC·  =·000045
Q$FO·  =·000044
Q$FP·  =·000046
Q$HBF·=·000002·
Q$ICP·=·000006
Q$IHB·=·000003
Q$IHRL=·000002·
Q$IMRP=·000007
Q$LBD·=·001000
Q$LBDP=·001001
Q$LBP·=·000001
Q$LDCD=·000003
Q$LDMD=·000004
Q$LDPP=·002000
Q$LHP·=·010000
Q$MNC·=·140000
Q$MR·  =·000052·
Q$MRP·=·000040
Q$MRP2=·000240
Q$MSC·=·040000
Q$MSET=·000004
Q$MSP·=·100000
Q$NCLK=·176000
Q$PP·  =·000100
Q$PPSW=·000320
Q$PP2·=·000300
Q$QHLT=·000013
Q$QL   =·000043
Q$QLA·=·000053
Q$QLB·=·000054
Q$QLR·=·000001
Q$QU   =·000042
Q$RDCD=·000005
Q$RDMD=·000006
Q$REBK=·001000
Q$RNC·=·006000
Q$RSC·=·004000
Q$RSET=·000010
Q$SM   =·100000
Q$SP   =·000120
Q$SP2·=·000340
RE3      000224RG·    002·
RE3C1    000310RG·    002·
RE3C2·   000316RG     002·
RE3IN·   000300R·     002·
RE3LB·   000324RG·    002·
RE3LN·=·000003 G·
RE3PUT·  000332R·     002·
RE3TBL·  000014RG·    002·
RE3X     000400R·     002·
RGQ.EM=·000200
RGQ.VA=·020000
RS3      000416RG·    002·
RTNPT·=·****** GX·
SCAN   =·****** GX·
SEQ.CI=·000010
ST3      000030RG·    002·
ST3C1    000150RG·    002·
ST3C2·   000160RG     002·
ST3IN·   000140R·     002·
ST3LB·   000170RG·    002·
ST3LN·=·000003 G·
ST3LP·   000200R·     002·
ST3TBL·  000000RG·    002·
ST3X     000214R·     002·
S$CLR·=·000000
S$LA   =·000001
S$QB   =·000005
S$QR   =·000006
S$QX   =·000004
S$SR   =·000007
S$Si   =·000010
S$S2   =·000014
TD$CTR=·176370
TD$CTW=·176360
TD$INL=·004000
TD$MEM=·000270
TD$OAR=·176344
TD$OTR=·176346
TD$QRD=·000274
TD$SW=·176376
TD$TAR=·176372
TD$TAW=·176362
TD$TDR=·176374
```

```
TD$TDW= 176364          T$RNB· =·000040         WORD23=·000056          WORD5 =·000012·         WORD76=·000230
T$AD·  =·000020         T$RSET=·040000          WORD24=·000060          WORD50=·000144          WORD77=·000232
T$BA·  =·000002·        T$SC·  =·000022·        WORD25=·000062·         WORD51=·000146          WORD78=·000234
T$BD·  =·000010         T$SCLK=·020000          WORD26=·000064          WORD52=·000150          WORD79=·000236
T$BSO· =·100000         T$SEG1=·000000          WORD27=·000066          WORD53=·000152·         WORD8 =·000020
T$BT·  =·000020         T$SEG2=·000001          WORD28=·000070          WORD54=·000154          WORD80=·000240
T$BTAR=·000030          T$SEG3=·000002·         WORD29=·000072·         WORD55=·000156·         WORD81=·000242
T$BTD· =·002000         T$SO·  =·000001         WORD3 =·000006          WORD56=·000160          WORD82=·000244
T$CD·  =·000100         T$UBUS=·100000          WORD30=·000074          WORD57=·000162·         WORD83=·000246
T$CLK· =·002000         T$1CLK=·000400          WORD31=·000076          WORD58=·000164          WORD84=·000250
T$DISK=·000200          T$8BEN=·000020          WORD32=·000100          WORD59=·000166          WORD85=·000252
T$DRD· =·000004         UBD.IN=·000020          WORD33=·000102·         WORD6 =·000014          WORD86=·000254
T$EMEM=·010000          UNPK·  =·****** GX·     WORD34=·000104          WORD60=·000170          WORD87=·000256
T$FSAA=·000000          WORD0 =·000000          WORD35=·000106          WORD61=·000172·         WORD88=·000260
T$FSAB=·000004          WORD1 =·000002·         WORD36=·000110          WORD62=·000174          WORD89=·000262
T$FSAC=·000014          WORD10=·000024          WORD37=·000112·         WORD63=·000176          WORD9 =·000022
T$FSB2=·000010          WORD11=·000026          WORD38=·000114          WORD64=·000200          WORD90=·000264
T$IB·  =·000026         WORD12=·000030          WORD39=·000116          WORD65=·000202·         WORD91=·000266
T$IBAR=·000024          WORD13=·000032·         WORD4 =·000010          WORD66=·000204          WORD92=·000270
T$IBE· =·020000         WORD14=·000034          WORD40=·000120          WORD67=·000206          WORD93=·000272
T$IBF· =·040000         WORD15=·000036          WORD41=·000122·         WORD68=·000210          WORD94=·000274
T$ICD· =·000040         WORD16=·000040          WORD42=·000124          WORD69=·000212·         WORD95=·000276
T$MODE=·004000          WORD17=·000042·         WORD43=·000126          WORD7 =·000016          WORD96=·000300
T$OB·  =·000036         WORD18=·000044          WORD44=·000130          WORD70=·000214          WORD97=·000302
T$OBE· =·004000         WORD19=·000046          WORD45=·000132·         WORD71=·000216          WORD98=·000304
T$OBF· =·010000         WORD2 =·000004          WORD46=·000134          WORD72=·000220          WORD99=·000306
T$OBRA=·000034          WORD20=·000050          WORD47=·000136          WORD73=·000222·         WRDVAL=·000310
T$OBWA=·000032·         WORD21=·000052·         WORD48=·000140          WORD74=·000224          XTREAD=·001000
T$OUTA=·100000          WORD22=·000054          WORD49=·000142·         WORD75=·000226          XTWRTE=·000400
T$RBDO=·000200

. ABS.  000000     000
        000000     001
BCREST· 000446     002·
ERRORS·DETECTED:  0

VIRTUAL·MEMORY·USED:  3196 WORDS·  (·13 PAGES)
DYNAMIC·MEMORY:  3860 WORDS·  ( 14 PAGES)
ELAPSED·TIME:  00:00:42·
BCREST,BCREST/-SP=[20,1]IM,[20,1]BCREST·
```

```
                        .TITLE· PPS·
                        .PSECT· PPS·
000000
                ;
                ;
                ;       HARDWARE·QUERY·RESOLVER·'MANUAL' DEBUGGING AIDS
                ;       PIPELINE·PROCESSORS·
                ;
                ;       PARSE·THE·COMMAND·LINE AND·TRANSFER·CONTROL·TO·ONE·OF·THE·
                ;       PPS·SUB-MODULES·OR·BACK·TO·QMAIN. CONTROL·COULD·HAVE BEEN·
                ;       PASSED·TO·PPS·IN·ONE·OF·THREE·WAYS:
                ;       1. FROM·QMAIN·IF·PPS·WAS·SELECTED·AS·THE·FIRST·PROCESSOR·
                ;          UPON·ENTRY·TO·THE·PROGRAM·
                ;          >PP·LD·QW·0
                ;       2. FROM·QMAIN·IF·PPS·WAS·SELECTED·FROM·ONE OF·QMAIN'S·
                ;          OTHER·SUB-MODULES·
                ;          CP>PP·LD·QW·0
                ;       3. ON·RETURN FROM·ONE·OF·PPS'S·SUB-MODULES
                ;          PP>LD·QW·0
                ;
                ;       PPS·PARSES·AT·THE·SECOND·LEVEL·OF·CONTROL·(SEE·NOTES AT·
                ;       QMAIN·ON·LEVELS·OF·CONTROL). IN·THE·EXAMPLES·ABOVE, PPS·
                ;       WOULD·PARSE·'LD' AND·TRANSFER·CONTROL·TO·THE·PPS·SUB-
                ;       MODULE·PPLD. IF·PPS·SHOULD·ENCOUNTER·A·STRING·WHICH·IS·
                ;       NOT·A·VALID·COMMAND·MNEMONIC, PPS·CONSIDERS·THE·STRING·TO·
                ;       BE A·PROCESSOR·MNEMONIC·AND·RETURNS·CONTROL·TO·QMAIN. EG:
                ;       PP>CP·LD·CS·0
                ;       THE·STRING·'CP' IS·NOT A·VALID·PPS·(SECOND LEVEL) COMMAND.
                ;       PPS·RETURNS·CONTROL·TO QMAIN·WHICH·IN·TURN WILL·TRANSFER·
                ;       CONTROL·TO·ITS·SUB-MODULE·CP.
                ;
                ;       PPS·SUB-MODULES:
                ;       PPLD·    LOAD MEMORIES·
                ;       PPPR·    PRINT·MEMORY·CONTENTS·
                ;       PPREST·  ALL·OTHER·PPS·COMMANDS·
                ;
                ;
                        .MCALL·  WTSE$S,CLEF$S·
```

```
                                        ;
                                        ;
                                        ;       TABLE·OF·PPS COMMAND·MNEMONICS·AND·THEIR·ASSOCIATED·
                                        ;       ROUTINE·ADDRESSES·
                                        ;
                                        ;
000000                                  PPTBL:
000000      123     124                         .ASCII·  /ST/                   ;STORE·INTO·A·REG·
000002·  000000G·                               .WORD·   ST4
000004      122·    105                         .ASCII·  /RE/                   ;READ·FROM A·REG·
000006   000000G·                               .WORD·   RE4
000010      114     104                         .ASCII·  /LD/                   ;LOAD·MEMORY·
000012·  000000G·                               .WORD·   LD4
000014      120     122                         .ASCII·  /PR/                   ;PRINT·FROM·MEMORY·
000016   000000G·                               .WORD·   PPPR4
000020      105     122                         .ASCII·  /ER/                   ;MEMORY·ERASE·
000022·  000000G·                               .WORD·   ER4
000024      120     123                         .ASCII·  /PS/                   ;PAGE·SWITCH·
000026   000000G·                               .WORD·   PS4
000030      103     114                         .ASCII·  /CL/                   ;CALL·QR·LOADER·
000032·  000000G·                               .WORD·   CL4
            000007              PPNUM·  =·      <.-PPTBL>/4
                                        ;
                                        ;
                                        ;       CODE·FOR·MEMORY·SELECTION·
                                        ;
000034   000000                 CODE::  .WORD·   0
```

```
                                  ;
                                  ;
                                  ;       PPS ROUTINES
                                  ;
                                  ;
000036                            PPS::
000036                                    CALL    FIND                   ;LOCATE THE COMMAND IN THE COMMAND LINE
000042  103003                            BCC     1$                     ;OK, CONTINUE
000044                                    CALL    ERR3
000050  000422                            BR      PPSXX
000052  022700  000002            1$:     CMP     #2,R0                  ;COMMANDS ARE 2 CHARS
000056  001403                            BEQ     2$
000060                                    CALL    ERR8
000064  000414                            BR      PPSXX                  ;TRY AGAIN
                                  ;
000066  012700  000007            2$:     MOV     #PPNUM,R0              ;R0 = NUMBER OF COMMANDS
000072  012702  000000'                   MOV     #PPTBL,R2              ;R2 -> TABLE OF COMMAND MNEMONICS
000076                                    CALL    SCAN                   ;FIND MATCH IN TABLE
000102  103003                            BCC     3$                     ;OK, CONTINUE
000104                                    CALL    ERR12                  ;COMMAND NOT IN TABLE
000110  000402                            BR      PPSXX                  ;TRY AGAIN
                                  ;
                                  ;       JUMP TO THE ROUTINE THAT GOVERNS THE COMMAND
                                  ;
000112  000171  000000            3$:     JMP     @(R1)
                                  ;
                                  ;
                                  ;       LOCAL PPS LOOP; INCLUDE PROMPT FOR PPS COMMAND;
                                  ;
                                  ;
000116                            PPSXX::
000116  012767  050120  000000C           MOV     #"PP,GCMBLK+G.DPRM+2   ;MOVE PPS NAME TO GCM BLOCK
000124                                    CALL    GCONLY                 ;PROMPT
000130                                    CALL    FIND                   ;LOCATE THE COMMAND IN THE COMMAND LINE
000134  103003                            BCC     1$                     ;OK, CONTINUE
000136                                    CALL    ERR3
000142  000765                            BR      PPSXX
000144  022700  000002            1$:     CMP     #2,R0                  ;COMMANDS ARE 2 CHARS
000150  001403                            BEQ     2$
000152                                    CALL    ERR8
000156  000757                            BR      PPSXX                  ;TRY AGAIN
                                  ;
000160  012700  000007            2$:     MOV     #PPNUM,R0              ;R0 = NUMBER OF COMMANDS
000164  012702  000000'                   MOV     #PPTBL,R2              ;R2 -> TABLE OF COMMAND MNEMONICS
000170                                    CALL    SCAN                   ;FIND MATCH IN TABLE
000174  103002                            BCC     3$                     ;OK, CONTINUE
000176  000167  000000G                   JMP     COMXX                  ;RETURN TO 'MAIN'. LOOK FOR PROCESSOR MNEMONIC
                                  ;
                                  ;       JUMP TO THE ROUTINE THAT GOVERNS THE COMMAND
                                  ;
000202  000171  000000            3$:     JMP     @(R1)
```

```
                                        ;
                                        ;
                                        ;       SET UP Q-REGISTER
                                        ;
                                        ;
000206                                  QREG::
000206  012746  000200                          MOV     #<PLR.EN>,-(SP)         ;BIT PATTERN FOR PLR RIGHT
000212                                          CALL    MRPCR                   ;WRITE TO MRP CONTROL REGISTER
000216  012746  000200                          MOV     #RGQ.EN,-(SP)           ;ENABLE Q-REG LOAD
000222                                          CALL    LBMRP                   ;SEND ENABLE TO MRP
000226  012746  000202                          MOV     #<PLR.EN+MMLEFT>,-(SP)  ;BIT PATTERN FOR PLR LEFT
000232                                          CALL    MRPCR                   ;WRITE TO MRP CONTROL REG
000236  012746  020000                          MOV     #RGQ.VAL,-(SP)          ;SET Q REG READY
000242                                          CALL    LBMRP                   ;SEND COMMAND TO MRP
000246  016646  000002                          MOV     2(SP),-(SP)             ;SEND QEX ADDRESS
000252                                          CALL    LBMRP
                                        ;
                                        ;       EXTRA CLOCK FOR PPS
                                        ;
000256  012746  001001                          MOV     #<Q$LBD+Q$LBP>,-(SP)    ;CLEAR DRIVE AND PULSE
000262  052716  000360                          BIS     #Q$CSEL,(SP)            ;CLEAR SELECTION BITS
000266  012746  176000                          MOV     #Q$NCLK,-(SP)           ;SET NO-CLOCKS
000272  052716  000300                          BIS     #Q$PP2,(SP)             ;SELECT PPS
000276                                          CALL    CSR1
                                        ;
000302  012746  006000                          MOV     #Q$RNC,-(SP)            ;CLEAR PPS, NO-CLOCK
000306  005046                                  CLR     -(SP)                   ;SELECT NOTHING
000310                                          CALL    CSR1
                                        ;
                                        ;       DE-SELECTION
                                        ;
000314  012746  001001                          MOV     #<Q$LBD+Q$LBP>,-(SP)    ;CLEAR DRIVE AND PULSE
000320  052716  000360                          BIS     #Q$CSEL,(SP)            ;CLEAR SELECTION BITS
000324  012746  176000                          MOV     #Q$NCLK,-(SP)           ;SET NO-CLOCKS
000330                                          CALL    CSR1
                                        ;
000334  011666  000002                          MOV     (SP),2(SP)              ;MOVE RETURN ADDRESS DOWN STACK
000340  005726                                  TST     (SP)+                   ;POINT TO RETURN ADDRESS
000342                                          RETURN
```

```
                                        ;
                                        ;
                                        ;
                                        ;       LOAD QCL POINTER.
                                        ;
                                        ;
000344                                  STOP::
000344  016667  000002  176424                MOV     2(SP),QR$LBR            ;MOVE POINTER WORD TO LOD BUS REG.
000352  012746  001001                       MOV     #<Q$LBD+Q$LBP>,-(SP)    ;CLEAR DRIVE AND PULSE.
000356  052716  000360                       BIS     #Q$CSEL,(SP)            ;CLEAR SELECTION BITS.
000362  012746  176000                       MOV     #Q$NCLK,-(SP)           ;SET NO-CLOCKS.
000366                                       CALL    CSR1
                                        ;
000372  005046                               CLR     -(SP)                   ;CLEAR NOTHING.
000374  012746  001300                       MOV     #<Q$PP2+Q$LBD>,-(SP)    ;SELECT PPS AND SET DRIVE.
000400                                       CALL    CSR1
                                        ;
                                        ;       SET FAL LOAD
                                        ;
000404  012767  004000  176422               MOV     #Q$FAL,QR$CR2           ;SET FAL LOAD.
                                        ;
                                        ;       EXTRA CLOCK FOR PPS.
                                        ;
000412  012746  000001                       MOV     #Q$LBP,-(SP)            ;CLEAR PULSE.
000416  052716  006000                       BIS     #Q$RNC,(SP)             ;CLEAR PPS NO-CLOCK.
000422  005046                               CLR     -(SP)                   ;SET NOTHING.
000424                                       CALL    CSR1
                                        ;
                                        ;       TURN OFF FAL LOAD.
                                        ;
000430  005067  176422                       CLR     QR$CR2
                                        ;
                                        ;       DE-SELECTION
                                        ;
000434  012746  001001                       MOV     #<Q$LBD+Q$LBP>,-(SP)    ;CLEAR DRIVE AND PULSE.
000440  052716  000360                       BIS     #Q$CSEL,(SP)            ;CLEAR SELECTION BITS.
000444  012746  176000                       MOV     #Q$NCLK,-(SP)           ;SET NO-CLOCKS.
000450                                       CALL    CSR1
                                        ;
000454  011666  000002                       MOV     (SP),2(SP)              ;MOVE RETURN ADDRESS DOWN STACK.
000460  005726                               TST     (SP)+                   ;POINT TO RETURN ADDRESS.
000462                                       RETURN
```

```
203                                     ;
204         000001                              .END·
```

```
ALUCKE=·040000          BYTE42=·000052·         BYTE94=·000136          MREN2·=·020000          Q$MSET=·000004,
ALUOE·=·004000          BYTE43=·000053          BYTE95=·000137          MRPCR·=·****** GX·      Q$MSP·=·100000
A01   =·010000          BYTE44=·000054          BYTE96=·000140          MSYN·=·000040,,,        Q$NCLK=·176000
BITVAL=·000000          BYTE45=·000055          BYTE97=·000141          N·   =·000144           Q$PP  =·000100
BIT0  =·000001          BYTE46=·000056          BYTE98=·000142·         PLB·=·000010            Q$PPSW=·000320
BIT1  =·000002·         BYTE47=·000057          BYTE99=·000143          PLC·=·000020            Q$PP2·=·000300
BIT10 =·002000          BYTE48=·000060          BYTVAL=·000144          PLD·=·000030            Q$QHLT=·000013
BIT11 =·004000          BYTE49=·000061          CBKALL=·001000          PLRWR·=·000200          Q$QL  =·000043
BIT12·=·010000          BYTE5 =·000005          CBKCLK=·000400          PLR.EN=·000200          Q$QLA·=·000053
BIT13 =·020000          BYTE50=·000062·         CL4   =·****** GX·      PPNUM·=·000007          Q$QLB·=·000054
BIT14 =·040000          BYTE51=·000063          CNOBRE=·100000          PPPR4·=·****** GX·      Q$QLR·=·000001
BIT15 =·100000          BYTE52=·000064          CODE·   000034RG·  002· PPS·   000036RG·  002· Q$QW  =·000042
BIT2·=·000004           BYTE53=·000065          COMXX·=·****** GX·      PPSXX·  000116RG·  002· Q$RDCD=·000005
BIT3  =·000010          BYTE54=·000066          CPCCEN=·010000          PPTBL·  000000R·   002· Q$RDMD=·000006
BIT4  =·000020          BYTE55=·000067          CPREAD=·040000          PS4   =·****** GX·      Q$REBK=·001000
BIT5  =·000040          BYTE56=·000070          CPWRTE=·020000          QREG·   000206RG·  002· Q$RNC·=·006000
BIT6  =·000100          BYTE57=·000071          CSADRD=·000004          QR$CR1=·176420          Q$RSC·=·004000
BIT7  =·000200          BYTE58=·000072·         CSEQCI=·100000          QR$CR2=·176422          Q$RSET=·000010
BIT8·=·000400           BYTE59=·000073          CSOE·=·000040           QR$LBR=·176424          Q$SM  =·100000
BIT9  =·001000          BYTE6 =·000006          CSR1  =·****** GX·      Q$ATTN=·000100          Q$SP  =·000120
BYTE0 =·000000          BYTE60=·000074          CSWRTE=·000100          Q$BCL·=·000001          Q$SP2·=·000340
BYTE1 =·000001          BYTE61=·000075          DBR.RD=·000001          Q$CCCP=·000040          RE4   =·****** GX·
BYTE10=·000012·         BYTE62=·000076          DB$CPP=·001457          Q$CHB·=·000400          RGO.EN=·000200
BYTE11=·000013          BYTE63=·000077          DB$SPT=·000026          Q$CHRL=·000200          RGO.VA=·020000
BYTE12=·000014          BYTE64=·000100          DB$TPC=·000023          Q$CLR·=·000040          SCAN  =·****** GX·
BYTE13=·000015          BYTE65=·000101          DISPGS=·100000          Q$CNC·=·030000          SEQ.CI=·000010
BYTE14=·000016          BYTE66=·000102·         DMAAWR=·000005          Q$CP·=·000060           STOP    000344RG·    002·
BYTE15=·000017          BYTE67=·000103          DMARRD=·000003          Q$CPCC=·000010          ST4   =·****** GX·
BYTE16=·000020          BYTE68=·000104          DMARWR=·000004          Q$CP2·=·000260          S$CLR·=·000000
BYTE17=·000021          BYTE69=·000105          ENBR·=·010000           Q$CSC·=·010000          S$LA  =·000001
BYTE18=·000022·         BYTE7 =·000007          ERR12·=·****** GX·      Q$CSEL=·000360          S$QB  =·000005
BYTE19=·000023          BYTE70=·000106          ERR3  =·****** GX·      Q$CSET=·000002·         S$QR  =·000006
BYTE2·=·000002·         BYTE71=·000107          ERR8  =·****** GX·      Q$CSP·=·020000          S$QX  =·000004
BYTE20=·000024          BYTE72=·000110          ER4   =·****** GX·      Q$DMA·=·000001          S$SR  =·000007
BYTE21=·000025          BYTE73=·000111          FIND·=·****** GX·       Q$ENBK=·040000          S$S1  =·000010
BYTE22=·000026          BYTE74=·000112·         GCMBLK=·****** GX·      Q$ENOP=·020000          S$S2  =·000014
BYTE23=·000027          BYTE75=·000113          GCONLY=·****** GX·      Q$FAL·=·004000          TD$CTR=·176370
BYTE24=·000030          BYTE76=·000114          G.DPRM=·****** GX·      Q$FC·=·000045           TD$CTW=·176360
BYTE25=·000031          BYTE77=·000115          LBMRP·=·****** GX·      Q$FO·=·000044           TD$INL=·004000
BYTE26=·000032·         BYTE78=·000116          LD4   =·****** GX·      Q$FP·=·000046           TD$MEM=·000270
BYTE27=·000033          BYTE79=·000117          LOC.EN=·000100          Q$HBF·=·000002·         TD$OAR=·176344
BYTE28=·000034          BYTE8 =·000010          LOC.WA=·040000          Q$ICP·=·000006          TD$OTR=·176346
BYTE29=·000035          BYTE80=·000120          LOC.WB=·100000          Q$IHB·=·000003          TD$ORD=·000274
BYTE3 =·000003          BYTE81=·000121          MAREN1=·000001          Q$IHRL=·000002·         TD$SW·=·176376
BYTE30=·000036          BYTE82=·000122·         MAREN2=·004000          Q$IMRP=·000007          TD$TAR=·176372
BYTE31=·000037          BYTE83=·000123          MARLOD=·010000          Q$LBD·=·001000          TD$TAW=·176362
BYTE32=·000040          BYTE84=·000124          MAROUT=·000002·         Q$LBDP=·001001          TD$TDR=·176374
BYTE33=·000041          BYTE85=·000125          MAR.LO=·002000          Q$LBP·=·000001          TD$TDW=·176364
BYTE34=·000042·         BYTE86=·000126          MAR.OU=·000040          Q$LDCD=·000003          T$AD  =·000020
BYTE35=·000043          BYTE87=·000127          MBKALL=·001000          Q$LDMD=·000004          T$BA  =·000002
BYTE36=·000044          BYTE88=·000130          MBKCLK=·000400          Q$LDPP=·002000          T$BD  =·000010
BYTE37=·000045          BYTE89=·000131          MMADRD=·000100          Q$LHP·=·010000          T$BSC =·100000
BYTE38=·000046          BYTE9 =·000011          MMLEFT=·000002·         Q$MNC·=·140000          T$BT  =·000020
BYTE39=·000047          BYTE90=·000132·         MMOE·=·000004           Q$MR·=·000052·          T$BTAR=·000030
BYTE4 =·000004          BYTE91=·000133          MMWRTE=·000010          Q$MRP·=·000040          T$BTD·=·002000
BYTE40=·000050          BYTE92=·000134          MNOBRE=·100000          Q$MRP2=·000240          T$CD  =·000100
BYTE41=·000051          BYTE93=·000135          MREN1·=·000001          Q$MSC·=·040000          T$CLK·=·002000
```

```
PPS     MACRO M1110  27-MAR-80 15:26  PAGE 10-2
SYMBOL TABLE
```

```
T$DISK= 000200          T$SO   = 000001          WORD29= 000072          WORD53= 000152          WORD78= 000234
T$DRD  = 000004          T$UBUS= 100000          WORD3  = 000006          WORD54= 000154          WORD79= 000236
T$EMEM= 010000          T$1CLK= 000400          WORD30= 000074          WORD55= 000156          WORD8  = 000020
T$FSAA= 000000          T$8BEN= 000020          WORD31= 000076          WORD56= 000160          WORD80= 000240
T$FSAB= 000004          UBD.IN= 000020          WORD32= 000100          WORD57= 000162          WORD81= 000242
T$FSAC= 000014          WORD0  = 000000          WORD33= 000102          WORD58= 000164          WORD82= 000244
T$FSB2= 000010          WORD1  = 000002          WORD34= 000104          WORD59= 000166          WORD83= 000246
T$IB   = 000026          WORD10= 000024          WORD35= 000106          WORD6  = 000014          WORD84= 000250
T$IBAR= 000024          WORD11= 000026          WORD36= 000110          WORD60= 000170          WORD85= 000252
T$IBE  = 020000          WORD12= 000030          WORD37= 000112          WORD61= 000172          WORD86= 000254
T$IBF  = 040000          WORD13= 000032          WORD38= 000114          WORD62= 000174          WORD87= 000256
T$ICD  = 000040          WORD14= 000034          WORD39= 000116          WORD63= 000176          WORD88= 000260
T$MODE= 004000          WORD15= 000036          WORD4  = 000010          WORD64= 000200          WORD89= 000262
T$OB   = 000036          WORD16= 000040          WORD40= 000120          WORD65= 000202          WORD9  = 000022
T$OBE  = 004000          WORD17= 000042          WORD41= 000122          WORD66= 000204          WORD90= 000264
T$OBF  = 010000          WORD18= 000044          WORD42= 000124          WORD67= 000206          WORD91= 000266
T$OBRA= 000034          WORD19= 000046          WORD43= 000126          WORD68= 000210          WORD92= 000270
T$OBWA= 000032          WORD2  = 000004          WORD44= 000130          WORD69= 000212          WORD93= 000272
T$OUTA= 100000          WORD20= 000050          WORD45= 000132          WORD7  = 000016          WORD94= 000274
T$RBDO= 000200          WORD21= 000052          WORD46= 000134          WORD70= 000214          WORD95= 000276
T$RNB  = 000040          WORD22= 000054          WORD47= 000136          WORD71= 000216          WORD96= 000300
T$RSET= 040000          WORD23= 000056          WORD48= 000140          WORD72= 000220          WORD97= 000302
T$SC   = 000022          WORD24= 000060          WORD49= 000142          WORD73= 000222          WORD98= 000304
T$SCLK= 020000          WORD25= 000062          WORD5  = 000012          WORD74= 000224          WORD99= 000306
T$SEG1= 000000          WORD26= 000064          WORD50= 000144          WORD75= 000226          WRDVAL= 000310
T$SEG2= 000001          WORD27= 000066          WORD51= 000146          WORD76= 000230          XTREAD= 001000
T$SEG3= 000002          WORD28= 000070          WORD52= 000150          WORD77= 000232          XTWRTE= 000400
```

```
. ABS.  000000     000
        000000     001
PPS     000464     002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3251 WORDS  ( 13 PAGES)
DYNAMIC MEMORY:  3860 WORDS  ( 14 PAGES)
ELAPSED TIME:  00:00:43
^PPS,PPS/-SP=[20,1]IM.[20,1]PPS
```

```
                                        .TITLE  PPLD
000000                                  .PSECT  PPLD
                                ;
                                ;
                                ;       HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
                                ;       PIPELINED PROCESSORS' TEST ROUTINES
                                ;
                                ;       COMMANDS:
                                ;       LD      LOAD PPS MEMORIES
                                ;
                                ;       ONCE A COMMAND HAS BBEN EXECUTED (OR AN ERROR ENCOUNTERED)
                                ;       THIS MODULE RETURNS CONTROL TO THE MODULE PPS AT LOCATION
                                ;       'PPSXX'.
                                ;
                                ;
                                ; **********************
                                ;
                                ;       NOTE:
                                ;       THE COMMANDS TO LOAD QEX MEMORIES CANNOT BE USED
                                ;       UNLESS MRP MICROPROGRAM MEMORY AND CP CONTROL STORE
                                ;       HAVE BEEN LOADED VIA THE 'LOADER' PROGRAM.  QEX LOADING
                                ;       MAKES USE OF CP DATA MEMORY - THE PREVIOUS CONTENTS
                                ;       OF THAT MEMORY ARE DESTROYED.
                                ;
                                ; **********************
                                ;
                                ;
                                ;
                                        .MCALL  WTSE$S,CLEF$S
                                ;
                                ;
                                ;       TABLE OF VALID PPS MEMORY MNEMONICS AND ADDRESSES OF THEIR
                                ;       ASSOCIATED ROUTINES
                                ;
000000                          LD4TBL::
000000      121     127                 .ASCII  /QW/                    ;QEX WINDOW MEMORY
000002  000322'                         .WORD   LD4QW
000004      121     114                 .ASCII  /QL/                    ;QEX LOCATION MEMORY
000006  000332'                         .WORD   LD4QL
000010      106     120                 .ASCII  /FP/                    ;FAL POINTER MEMORY
000012  000662'                         .WORD   LD4FP
000014      106     103                 .ASCII  /FC/                    ;FAL COUNTER MEMORY
000016  000672'                         .WORD   LD4FC
000020      121     122                 .ASCII  /QR/                    ;QLB REFERENCE PAGE
000022  001164'                         .WORD   LD4QR
000024      121     060                 .ASCII  /Q0/                    ;QLB PAGE 0
000026  001450'                         .WORD   LD4Q0
000030      121     061                 .ASCII  /Q1/                    ;QLB PAGE 1
000032  001456'                         .WORD   LD4Q1
000034      121     062                 .ASCII  /Q2/                    ;QLB PAGE 2
000036  001464'                         .WORD   LD4Q2
        000010                  LD4LN   ==      <.-LD4TBL>/4
                                ;
                                ;
000040  000002                  CDADD:  .WORD   2                       ;LOAD QEX ADDR/DATA INTO CD
000042  000000                  QXCNT:  .WORD   0                       ;NUMBER OF QEX ADDR/DATA PAIRS
```

```
 58 000044  000000                  CDDAT:  .WORD·  0                       ;DATA·FIELD·FOR·TRANSFER·TO·CD
 59 000046  000000                  CODE:   .WORD·  0                       ;CODE·FOR·MEMORY·SELECTION·
 60                                 ;
 61                                 ;
 62·                                ;       LOAD·
 63                                 ;       PERFORM·THIRD·LEVEL·PARSING·
 64                                 ;       EG,·IN·THE·COMMAND:
 65                                 ;       PP>LD·QW·0
 66                                 ;       PARSE·THE·'QW'
 67                                 ;
 68                                 ;
 69 000050                          LD4::
 70 000050                                  CALL·   FIND                    ;LOCATE·MEMORY·MNEMONIC·IN·COMMAND·LINE·
 71 000054  103004                          BCC·    1$                      ;OK,·CONTINUE·
 72·000056                                  CALL·   ERR4                    ;'MISSING·OPERAND'
 73 000062· 000167  001672·                 JMP·    LD4X                    ;EXIT·
 74                                 ;
 75                                 ;       MATCH·THE·MNEMONIC·IN·THE·COMMAND·LINE·AGAINST·THE·TABLE·
 76                                 ;       OF·VALID·MNEMONICS·
 77                                 ;
 78 000066  012700  000010·         1$:     MOV·    #LD4LN,R0               ;NUMBER·OF·TABLE·ENTRIES·
 79 000072· 012702· 000000'                 MOV·    #LD4TBL,R2·             ;POINT·TO·TABLE·
 80 000076                                  CALL·   SCAN                    ;MATCH·AGAINST·COMMAND·LINE·
 81 000102· 103004                          BCC·    2$                      ;MATCH·WAS·MADE·
 82·000104                                  CALL·   ERR7                    ;'INVALID·MEMORY·MNEMONIC'
 83 000110  000167  001644                  JMP·    LD4X
 84                                 ;
 85                                 ;       SAVE·POINTER·TO·ROUTINE·ASSOCIATED·WITH·THE·MEMORY·MNEMONIC·
 86                                 ;       SCAN·THE·REMAINDER·OF·THE·COMMAND·LINE·FOR·CONTROL·INFORMATION:
 87                                 ;
 88                                 ;       1. START·ADDRESS·ONLY,·REPEAT·PROMPT·FOR·EACH·MEMORY·LOCATION·
 89                                 ;          FROM·THE·START·ADDRESS·FORWARD.·NB.·END·OF·MEMORY·(AS·DETECTED·
 90                                 ;          BY·THE·QMAIN·SUBROUTINE·PDATA)·OR·A·<CR>·RESPONSE·TO·THE·PROMPT·
 91                                 ;          WILL·TERMINATE·THE·LOAD.
 92·                                ;          PP>LD·QW·0
 93                                 ;       2. START·ADDRESS,·LOOP·INDICATOR.·LOOP·ON·THE·LOADING·OF·
 94                                 ;          THIS·ONE·MEMORY·LOCATION·ONLY.
 95                                 ;          PP>LD·QW·0·L·
 96                                 ;       3. START·ADDRESS,·END·ADDRESS,·NO·LOOP,·ISSUE·ONLY·ONE·
 97                                 ;          PROMPT·AND·FILL·MEMORY·(BETWEEN·START·AND·END·ADDRESSES)·
 98                                 ;          WITH·THIS·VALUE.
 99                                 ;          PP>LD·QW·0·7
100                                 ;       4. START·ADDRESS,·END·ADDRESS,·LOOP·INDICATOR,·ISSUE·ONLY·ONE·
101                                 ;          PROMPT·AND·LOOP·ON·THE·LOADING·OF·MEMORY·(BETWEEN·START·
102·                                ;          AND·END·ADDRESSES)·WITH·THIS·VALUE.
103                                 ;          PP>LD·QW·0·7·L·
104                                 ;
105                                 ;
106 000114  010167  000000G·        2$:     MOV·    R1,RTNPT·               ;SAVE·POINTER·
107 000120                                 CALL·   FIND                    ;LOCATE·START·ADDRESS·IN·COMMAND·LINE·
108 000124  103004                         BCC·    3$                      ;OK,·CONTINUE·
109 000126                                 CALL·   ERR4                    ;'MISSING·OPERAND'
110 000132· 000167  001622·                JMP·    LD4X                    ;EXIT·
111 000136                         3$:     CALL·   PACK                    ;CONVERT·COMMAND·LINE·VALUE·TO·BINARY·
112·000142· 103004                         BCC·    4$                      ;CONVERSION·SUCCESSFUL·
113 000144                                 CALL·   ERR5                    ;'INVALID·NUMERIC·VALUE'
114 000150  000167  001604                 JMP·    LD4X
```

```
                                        ;
000154  016767  000000G 000000G 4$:     MOV     BINWD,MSTRT             ;SAVE LOADING START ADDRESS
000162  016767  000000G 000000G        MOV     BINWD,MSTR2             ;SAVE IT TWICE (FOR REFRESH ON LOOP)
000170  012767  177777  000000G        MOV     #-1,MEND                ;INIT END ADDRESS
                                        ;
                                        ;       START ADDRESS HAS BEEN FOUND, SCAN FOR END ADDRESS OR LOOP
                                        ;       INDICATOR (CONDITIONS 2, 3 ABOVE).  IF THERE IS NOTHING
                                        ;       FURTHER IN THE COMMAND LINE, CONDITION 1 IS IN EFFECT.
                                        ;
000176
000202  103004                          CALL    FIND                    ;SCAN COMMAND LINE
000204  052767  000000G 000000G        BCC     5$                      ;SOMETHING THERE
000212  000437                         BIS     #RP,BASE                ;SIGNAL TO REPEAT PROMPT
                                        BR       9$                     ;JUMP TO RTN
                                        ;
000214  122711  000114          5$:    CMPB    #'L,(R1)                ;LOOP INDICATOR
000220  001006                         BNE     6$                      ;NO, MUST BE UPPER ADDRESS
000222  016767  000000G 000000G        MOV     MSTRT,MEND              ;SET END ADDR = START ADDR
000230                                  CALL    HANG                    ;HOW TO STOP LOOP
000234  000426                         BR      9$                      ;JUMP TO RTN
                                        ;
000236                          6$:     CALL    PACK                    ;CONVERT UPPER ADDRESS
000242  103004                         BCC     7$                      ;OK, CONTINUE
000244                                  CALL    ERR5                    ;INVALID NUMERIC
000250  000167  001504                 JMP     LD4X                    ;EXIT
                                        ;
                                        ;       SAVE END ADDRESS (BINARY)
                                        ;       CHECK FOR LOOP INDICATOR AFTER END ADDRESS (CONDITION 4)
                                        ;
000254  016767  000000G 000000G 7$:     MOV     BINWD,MEND              ;SET UP END ADDRESS
000262                                  CALL    FIND                    ;CHECK FOR LOOP INDICATOR
000266  103411                         BCS     9$                      ;NO LOOP
000270  122711  000114                 CMPB    #'L,(R1)                ;CORRECT INDICATOR
000274  001404                         BEQ     8$                      ;YES, CONTINUE
000276                                  CALL    ERR11           ;LOOP OPTION ERROR
000302  000167  001452                 JMP     LD4X
000306                          8$:     CALL    HANG                    ;HOW TO STOP LOOP
                                        ;
000312  016701  000000G         9$:     MOV     RTNPT,R1                ;POINT TO ROUTINE
000316  000171  000000                 JMP     @(R1)                   ;JUMP TO ROUTINE
                                        ;
                                        ;
                                        ;       LOAD QEX WINDOW MEMORY
                                        ;       LOAD QEX LOCATION MEMORY
                                        ;
                                        ;
000322  012767  000042  177516  LD4QW:: MOV     #Q$QW,CODE              ;SET MEMORY SELECT CODE = WINDOW
000330  000403                         BR      QEX
000332  012767  000043  177506  LD4QL:: MOV     #Q$QL,CODE              ;SET MEMORY SELECT CODE = LOCATION
                                        ;
000340  016746  000000G         QEX:    MOV     QXHIGH,-(SP)            ;SUPPLY UPPER MEMORY LIMIT
000344  016746  000000G                 MOV     QXLOW,-(SP)             ;LOWER LIMIT
000350                                  CALL    BUFSET                  ;PREPARE FOR LOAD
000354  103002                         BCC     1$                      ;OK, CONTINUE
000356  000167  001376                 JMP     LD4X                    ;ERROR, EXIT
                                        ;
                                        ;       WRITE DATA FOR QEX INTO CP DATA MEMORY, SKIP CD WORDS 0, 1
                                        ;       (RESERVED FOR WRITE FLAG AND TRANSFER COUNT - SEE BELOW).
```

```
                                ;       WRITE QEX DATA IN THE FORM ADDR,DATA,ADDR,DATA ...
                                ;       WHEN THE LOADING OF CD IS COMPLETED, CALLL THE SUBROUTINE
                                ;       QXLOADD WHICH IN TURN STARTS UP THE MICROCODE TO LOAD THE
                                ;       QEX.
                                ;
000362  012767  000002  177450  1$:     MOV     #2,CDADD                ;WRITE INTO CD STARTING AT LOC 2
000370  005067  177446                  CLR     QXCNT                   ;CLEAR QEX ADDR/DATA PAIRS COUNT
000374  005046                          CLR     -(SP)                   ;START MICROCODE AT 0
000376                                  CALL    SEQCS
000402  005046                          CLR     -(SP)                   ;RESET BR INHIBIT
000404                                  CALL    CPCR
000410  012746  000377                  MOV     #377,-(SP)              ;SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
000414                                  CALL    SEQMM
000420  005046                          CLR     -(SP)                   ;RESET BR INHIBIT
000422                                  CALL    MRPCR
000426  012767  001000  176422          MOV     #Q$REBK,QR$CR2          ;RE-ARM INTERRUPTS
000434  012767  120000  176422          MOV     #<Q$SM+Q$ENOP>,QR$CR2   ;SET SEARCH MODE + ENABLE INTERRUPTS
000442  012746  000360                  MOV     #Q$CSEL,-(SP)           ;CLEAR ALL SELECTIONS
000446  052716  001001                  BIS     #<Q$LBD+Q$LBP>,(SP)     ;CLEAR DRIVE AND PULSE
000452  052716  030000                  BIS     #Q$CNC,(SP)             ;CLEAR CP NO-CLOCK
000456  005046                          CLR     -(SP)                   ;SET NOTHING
000460                                  CALL    CSR1
                                ;
000464  032767  000000G 000000G         BIT     #RP,BASE                ;REPEAT PROMPT ?
000472  001430                          BEQ     4$                      ;NO
000474                          2$:     CALL    PDATA                   ;READ DATA FROM COMMAND LINE
000500  103421                          BCS     3$                      ;END OF MEMORY - LOAD QEX
000502  102420                          BVS     3$                      ;<CR> RESPONSE - LOAD QEX
000504  016767  000000G 177332          MOV     MSTR2,CDDAT             ;WRITE QEX ADDR TO CP DATA MEMORY
000512                                  CALL    LOADCD                  ;LOAD ONE WORD
000516  016767  000000G 177320          MOV     DATA1,CDDAT             ;WRITE QEX DATA TO CP DATA MEMORY
000524                                  CALL    LOADCD
000530  005267  177306                  INC     QXCNT                   ;COUNT NUMBER OF QEX ADDR/DATA PAIRS
000534  066767  000000G 000000G         ADD     INCVAL,MSTR2            ;BUMP LOAD ADDRESS
000542  000754                          BR      2$                      ;REPEAT
                                ;
000544                          3$:     CALL    LOADQX                  ;LOAD QEX
000550  000167  001204                  JMP     LD4X                    ;EXIT
                                ;
                                ;       PROMPT ONCE THEN FILL MEMORY
                                ;
000554                          4$:     CALL    PDATA                   ;PROMPT
000560  103401                          BCS     5$                      ;END OF MEMORY OR ERROR
000562  102004                          BVC     6$                      ;NORMAL CONTINUATION
000564                          5$:     CALL    LOADQX                  ;LOAD QEX
000570  000167  001164                  JMP     LD4X
000574  016767  000000G 177242  6$:     MOV     MSTR2,CDDAT             ;WRITE QEX ADDR TO CP DATA MEMORY
000602                                  CALL    LOADCD                  ;LOAD ONE WORD
000606  016767  000000G 177230          MOV     DATA1,CDDAT             ;WRITE QEX DATA TO CP DATA MEMORY
000614                                  CALL    LOADCD
000620  005267  177216                  INC     QXCNT                   ;INC QEX ADDR/DATA PAIRS COUNT
000624  066767  000000G 000000G         ADD     INCVAL,MSTR2            ;ADVANCE ADDRESS
000632  026767  000000G 000000G         CMP     MSTR2,MEND              ;HAS UPPER MEMORY LIMIT BEEN REACHED
000640  101755                          BLOS    6$                      ;NO, CONTINUE
000642                          7$:     CALL    LOADQX                  ;LOAD QEX
000646  032767  000000G 000000G         BIT     #LOOP,BASE              ;LOOP ON ?
000654  001372                          BNE     7$                      ;YES, CONTINUE
```

```
  229 000656  000167  001076                  JMP     LD4X                    ;ELSE EXIT
  230                                 ;
  231                                 ;
  232                                 ;       FAL POINTER MEMORY.
  233                                 ;       FAL COUNTER MEMORY.
  234                                 ;
  235                                 ;
  236 000662  012767  000046  177156  LD4FP:: MOV     #Q$FP,CODE              ;SET MEMORY SELECT CODE = POINTER.
  237 000670  000403                  BR      FAL
  238 000672  012767  000045  177146  LD4FC:: MOV     #Q$FC,CODE              ;SET MEMORY SELECT CODE = COUNTER.
  239                                 ;
  240 000700  016746  000000G         FAL:    MOV     FAHIGH,-(SP)            ;SUPPLY UPPER MEMORY LIMIT.
  241 000704  016746  000000G                 MOV     FALOW,-(SP)             ;LOWER LIMIT.
  242 000710                          CALL    BUFSET                          ;PREPARE FOR LOAD.
  243 000714  103002                  BCC     1$                              ;OK, CONTINUE.
  244 000716  000167  001036          JMP     LD4X                            ;ERROR, EXIT.
  245                                 ;
  246 000722  032767  000000G 000000G 1$:     BIT     #RP,BASE                ;REPEAT PROMPT.
  247 000730  001436                  BEQ     5$                              ;NO, ONCE ONLY.
  248 000732                          2$:     CALL    PDATA                   ;READ DATA FROM COMMAND LINE.
  249 000736  103401                  BCS     3$                              ;END OF MEMORY.
  250 000740  102006                  BVC     4$                              ;NO <CR> RESPONSE, CONTINUE.
  251 000742  012746  077777          3$:     MOV     #077777,-(SP)           ;VALUE FOR QCL POINTER.
  252 000746                          CALL    STQP                            ;LOAD QCL POINTER.
  253 000752  000167  001002          JMP     LD4X
  254                                 ;
  255 000756  016746  000000G         4$:     MOV     MSTR2,-(SP)             ;LOAD ADDR INTO QCL POINTER.
  256 000762                          CALL    STQP
  257 000766  016746  177054          MOV     CODE,-(SP)                      ;SELECT MEMORY.
  258 000772                          CALL    PPCR                            ;WRITE SELECTION TO CONTROL REG.
  259 000776  016746  000000G         MOV     DATA1,-(SP)                     ;SEND DATA WORD TO FAL MEMORY.
  260 001002                          CALL    LBPP
  261 001006  012746  000040          MOV     #Q$CLR,-(SP)
  262 001012                          CALL    PPCR
  263                                 ;
  264 001016  066767  000000G 000000G         ADD     INCVAL,MSTR2            ;BUMP ADDRESS.
  265 001024  000742                  BR      2$                              ;REPEAT.
  266                                 ;
  267                                 ;       PROMPT ONCE THEN FILL MEMORY.
  268                                 ;
  269 001026                          5$:     CALL    PDATA                   ;PROMPT.
  270 001032  103401                  BCS     6$                              ;END OF MEMORY.
  271 001034  102006                  BVC     7$                              ;NO <CR> RESPONSE, CONTINUE.
  272 001036  012746  077777          6$:     MOV     #077777,-(SP)           ;VALUE FOR QCL POINTER.
  273 001042                          CALL    STQP                            ;LOAD QCL POINTER.
  274 001046  000167  000706          JMP     LD4X
  275                                 ;
  276 001052  016746  000000G         7$:     MOV     MSTR2,-(SP)             ;LOAD ADDR INTO QCL POINTER.
  277 001056                          CALL    STQP
  278 001062  016746  176760          MOV     CODE,-(SP)                      ;SELECT MEMORY.
  279 001066                          CALL    PPCR                            ;WRITE SELECTION TO CONTROL REG.
  280 001072  016746  000000G         MOV     DATA1,-(SP)                     ;SEND DATA WORD TO FAL MEMORY.
  281 001076                          CALL    LBPP
  282 001102  012746  000040          MOV     #Q$CLR,-(SP)                    ;CLEAR PPS
  283 001106                          CALL    PPCR
  284                                 ;
  285 001112  066767  000000G 000000G         ADD     INCVAL,MSTR2            ;ADVANCE ADDRESS.
```

```
286 001120  026767  000000G·000000G·        CMP·    MSTR2,MEND·             ;HAS·UPPER·MEMORY·LIMIT·BEEN·REACHED·
287 001126  101751                         BLOS·   7$                      ;NO, CONTINUE·
288 001130  032767  000000G·000000G·        BIT·    #LOOP,BASE·             ;LOOP·ON·
289 001136  001006                         BNE·    8$                      ;YES, CONTINUE·
290 001140  012746  077777                 MOV·    #077777,-(SP)           ;VALUE·FOR·QCL·POINTER·
291 001144                                 CALL·   STQP                    ;LOAD·QCL·POINTER·
292·001150  000167  000604                 JMP·    LD4X
293 001154  016767  000000G·000000G·8$:    MOV·    MSTRT,MSTR2·            ;REINITIALIZE·ADDRESS·
294 001162·000733                          BR      7$
295                                 ;
296                                 ;
297                                 ;       LOAD·QLB·REFERENCE·PAGE·
298                                 ;
299                                 ;
300 001164                          LD4QR::
301 001164  016746  000000G·               MOV·    LHHIGH,-(SP)            ;SUPPLY·UPPER·MEMORY LIMIT·
302·001170  016746  000000G·               MOV·    LHLOW,-(SP)             ;LOWER·LIMIT·
303 001174                                 CALL·   BUFSET·                 ;PREPARE·FOR·LOAD·
304 001200  103002·                        BCC·    1$                      ;OK, CONTINUE·
305 001202·000167  000552                  JMP·    LD4X                    ;ERROR, EXIT·
306                                 ;
307 001206  032767  000000G·000000G·1$:    BIT·    #RP,BASE·               ;REPEAT·PROMPT·
308 001214  001436                         BEQ·    5$                      ;NO, ONCE·ONLY·
309 001216                          2$:    CALL·   PDATA·                  ;READ·DATA FROM·COMMAND·LINE·
310 001222·103401                          BCS·    3$                      ;END·OF·MEMORY·
311 001224  102006                         BVC·    4$                      ;NO·<CR> RESPONSE, CONTINUE·
312·001226  012746  000013          3$:    MOV·    #Q$QHLT,-(SP)           ;HALT·CODE
313 001232·                                CALL·   PPCR
314 001236  000167  000516                 JMP·    LD4X
315                                 ;
316 001242·012746·000053            4$:    MOV·    #Q$QLA,-(SP)            ;ADDRESS·SELECT·FOR·QLB·PAGE·
317 001246                                 CALL·   PPCR                    ;SEND·TO·PP·CONTROL·REG·
318 001252·016746  000000G·                MOV·    MSTR2,-(SP)             ;ACTUAL·ADDRESS·
319 001256                                 CALL·   LBPP                    ;SEND·TO·PP·
320 001262·012746  000001                  MOV·    #Q$QLR,-(SP)            ;SELECT·QLB·REF·MEMORY·
321 001266                                 CALL·   PPCR
322·001272·016746  000000G·                MOV·    DATA1,-(SP)             ;DATA·WORD FOR·MEMORY·
323 001276                                 CALL·   LBPP                    ;SEND·DATA TO·PPS·
324                                 ;
325 001302·066767  000000G·000000G·        ADD·    INCVAL,MSTR2·           ;BUMP·ADDRESS·
326 001310  000742·                        BR      2$                      ;REPEAT·
327                                 ;
328·                                ;       PROMPT·ONCE·THEN·FILL·MEMORY·
329                                 ;
330 001312·                         5$:    CALL·   PDATA·                  ;PROMPT·
331 001316·103401                          BCS·    6$                      ;END·OF·MEMORY·
332·001320  102006                         BVC·    7$                      ;NO·<CR> RESPONSE, CONTINUE·
333 001322·012746  000013          6$:    MOV·    #Q$QHLT,-(SP)           ;HALT·CODE
334 001326                                 CALL·   PPCR
335 001332·000167  000422                  JMP·    LD4X
336                                 ;
337 001336  012746  000053          7$:    MOV·    #Q$QLA,-(SP)            ;ADDRESS·SELECT·FOR·QLB·PAGE·
338 001342·                                CALL·   PPCR                    ;SEND·TO·PP·CONTROL·REG·
339 001346  016746  000000G·               MOV·    MSTR2,-(SP)             ;ACTUAL·ADDRESS·
340 001352·                                CALL·   LBPP                    ;SEND·TO·PP·
341 001356  012746  000001                 MOV·    #Q$QLR,-(SP)            ;SELECT·QLB·REF·MEMORY·
342·001362·                                CALL·   PPCR
```

```
343 001366  016746  000000G·····            MOV·    DATA1,-(SP)          ;DATA·WORD FOR·MEMORY·
344 001372·                                 CALL·   LBPP                 ;SEND·DATA TO·PPS·
345                                 ;
346 001376  066767  000000G·000000G·        ADD·    INCVAL,MSTR2·        ;ADVANCE·ADDRESS·
347 001404  026767  000000G·000000G·        CMP·    MSTR2,MEND·          ;HAS·UPPER MEMORY·LIMIT·BEEN·REACHED·
348 001412· 101751                          BLOS·   7$                   ;NO, CONTINUE·
349 001414  032767  000000G·000000G·        BIT·    #LOOP,BASE·          ;LOOP·ON·
350 001422· 001006                          BNE·    8$                   ;YES, CONTINUE·
351 001424  012746  000013                  MOV·    #Q$QHLT,-(SP)        ;HALT·CODE
352·001430                                  CALL·   PPCR                 ;WRITE·TO·PPS·CONTROL·REGISTER
353 001434  000167  000320                  JMP·    LD4X
354 001440  016767  000000G·000000G·8$:    MOV·    MSTRT,MSTR2·         ;REINITIALIZE·ADDRESS·
355 001446  000733                          BR      7$
356                                 ;
357                                 ;
358                                 ;       LOAD·QLB·PAGES·
359                                 ;
360                                 ;
361 001450                          LD4Q0::
362·001450  012746  000000                  MOV·    #0,-(SP)             ;SELECT PAGE·0
363 001454  000405                          BR      PGSEL·
364 001456                          LD4Q1::
365 001456  012746  000001                  MOV·    #1,-(SP)             ;SELECT·PAGE·1
366 001462· 000402·                         BR      PGSEL·
367 001464                          LD4Q2::
368 001464  012746  000002                  MOV·    #2,-(SP)             ;SELECT PAGE·2·
369                                 ;
370 001470                          PGSEL:· CALL·   SELPG·               ;SELECT·A·QLB·PAGE·
371 001474  016746  000000G·                MOV·    LHHIGH,-(SP)         ;SUPPLY·UPPER·MEMORY LIMIT·
372·001500  016746  000000G·                MOV·    LHLOW,-(SP)          ;LOWER·LIMIT·
373 001504                                  CALL·   BUFSET·              ;PREPARE·FOR·LOAD·
374 001510  103002·                         BCC·    1$                   ;OK, CONTINUE·
375 001512· 000167  000242                  JMP·    LD4X                 ;ERROR, EXIT·
376                                 ;
377 001516  032767  000000G·000000G· 1$:    BIT·    #RP,BASE·            ;REPEAT·PROMPT·
378 001524  001436                          BEQ·    5$                   ;NO, ONCE·ONLY·
379 001526                          2$:     CALL·   PDATA·               ;READ·DATA FROM·COMMAND·LINE·
380 001532· 103401                          BCS·    3$                   ;END·OF·MEMORY·
381 001534  102006                          BVC·    4$                   ;NO·<CR> RESPONSE, CONTINUE·
382·001536  012746  000013          3$:     MOV·    #Q$QHLT,-(SP)        ;HALT·CODE
383 001542·                                 CALL·   PPCR                 ;WRITE·TO·PPS·CONTROL·REGISTER
384 001546  000167  000206                  JMP·    LD4X
385                                 ;
386 001552· 012746  000053          4$:     MOV·    #Q$QLA,-(SP)         ;ADDRESS·SELECT·FOR·QLB·PAGE·
387 001556                                  CALL·   PPCR                 ;SEND·TO·PP·CONTROL·REG·
388 001562· 016746  000000G·                MOV·    MSTR2,-(SP)          ;ACTUAL·ADDRESS·
389 001566                                  CALL·   LBPP                 ;SEND·TO·PP·
390 001572· 012746  000054                  MOV·    #Q$QLB,-(SP)         ;SELECT·QLB·REF·MEMORY·
391 001576                                  CALL·   PPCR
392·001602· 016746  000000G·                MOV·    DATA1,-(SP)          ;DATA·WORD FOR·MEMORY·
393 001606                                  CALL·   LBPP                 ;SEND·DATA TO·PPS·
394                                 ;
395 001612· 066767  000000G·000000G·        ADD·    INCVAL,MSTR2·        ;BUMP·ADDRESS·
396 001620  000742·                         BR      2$                   ;REPEAT
397                                 ;
398                                 ;       PROMPT·ONCE·THEN·FILL·MEMORY·
399                                 ;
```

```
400 001622·                                5$:     CALL·   PDATA·          ;PROMPT·
401 001626  103401                                 BCS·    6$              ;END·OF·MEMORY·
402·001630  102006                                 BVC·    7$              ;NO, CR> RESPONSE, CONTINUE·
403 001632· 012746  000013·            6$:     MOV·    #Q$QHLT,-(SP)       ;HALT·CODE
404 001636                                         CALL·   PPCR            ;WRITE·TO·PPS·CONTROL·REGISTER
405 001642· 000167  000112                         JMP·    LD4X
406                                        ;
407 001646  012746  000053             7$:     MOV·    #Q$QLA,-(SP)        ;ADDRESS·SELECT·FOR·QLB·PAGE·
408 001652·                                        CALL·   PPCR            ;SEND·TO·PP·CONTROL·REG·
409 001656  016746  000000G·                       MOV·    MSTR2,-(SP)     ;ACTUAL·ADDRESS·
410 001662·                                        CALL·   LBPP            ;SEND·TO·PP·
411 001666  012746  000054                         MOV·    #Q$QLB,-(SP)    ;SELECT·QLB·MEMORY·
412·001672·                                        CALL·   PPCR
413 001676  016746  000000G·                       MOV·    DATA1,-(SP)     ;DATA·WORD·FOR·MEMORY·
414 001702·                                        CALL·   LBPP            ;SEND·DATA·TO·PPS·
415                                        ;
416 001706  066767  000000G·000000G·               ADD·    INCVAL,MSTR2·   ;ADVANCE·ADDRESS·
417 001714  026767  000000G·000000G·               CMP·    MSTR2,MEND·     ;HAS·UPPER·MEMORY·LIMIT·BEEN·REACHED·
418 001722· 101751                                 BLOS·   7$              ;NO, CONTINUE·
419 001724  032767  000000G·000000G·               BIT·    #LOOP,BASE·     ;LOOP·ON·
420 001732· 001006                                 BNE·    8$              ;YES, CONTINUE·
421 001734  012746  000013                         MOV·    #Q$QHLT,-(SP)   ;HALT·CODE
422·001740                                         CALL·   PPCR            ;WRITE·TO·PPS·CONTROL·REGISTER
423 001744  000167  000010                         JMP·    LD4X
424 001750  016767  000000G·000000G·       8$:     MOV·    MSTRT,MSTR2·    ;REINITIALIZE·ADDRESS·
425 001756  000733                                 BR      7$
426                                        ;
427                                        ;
428·001760                                 LD4X:
429 001760  012746  000040                         MOV·    #Q$CLR,-(SP)    ;CLEAR·PPS
430 001764                                         CALL·   PPCR            ;WRITE·TO·CONTROL·REG·
431 001770  042767  000000G·000000G·               BIC·    #RP,BASE·       ;CLEAR·PROMPT·FLAG·
432·001776                                         CALL·   KILL            ;KILL·AST·(IF·THERE·WAS·ONE)
433 002002· 000167  000000G·                       JMP·    PPSXX·
```

```
                                ;
                                ;
                                ;       LOAD QEX MEMORY
                                ;
                                ;
                                ;       FIRST WRITE THE FLAG ('1' = WRITE) AND THE TRANSFER COUNT
                                ;       INTO THE FIRST TWO LOCATIONS IN CP DATA MEMORY.
                                ;
002006                          LOADQX:
002006  005767  176030                  TST     QXCNT                   ;ANYTHING TO LOAD
002012  001507                          BEQ     1$                      ;NO, SHUT DOWN AND EXIT
                                ;
002014  005067  176020                  CLR     CDADD                   ;WRITE FLAG TO CD ADDR ZERO
002020  012767  000001  176016          MOV     #1,CDDAT                ;FLAG FOR 'WRITE QEX'
002026                                  CALL    LOADCD
002032  016767  176004  176004          MOV     QXCNT,CDDAT             ;WRITE QEX ADDR/DATA PAIRS COUNT TO CD ADDR 1
002040                                  CALL    LOADCD
                                ;
002044  005046                          CLR     -(SP)                   ;CLEAR NOTHING IN CSR1
002046  012746  176000                  MOV     #Q$NCLK,-(SP)           ;SET NO-CLOCKS
002052                                  CALL    CSR1
002056  005067  176422                  CLR     QR$CR2                  ;SET LOAD MODE
                                ;
                                ;       SELECT MEMORY TO BE LOADED (WINDOW OR LOCATION).
                                ;       START THE MICROCODE THAT DOES THE LOADING OF THE QEX.
                                ;
002062  016746  175760                  MOV     CODE,-(SP)              ;SELECT WINDOW OR LOCATION MEMORY
002066                                  CALL    PPCR
002072  012746  001760                  MOV     #1760,-(SP)             ;START CP MICROCODE AT X'3F0'
002076                                  CALL    SEQCS
002102  005046                          CLR     -(SP)                   ;RESET BR INHIBIT
002104                                  CALL    CPCR
002110  012746  000377                  MOV     #377,-(SP)              ;SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
002114                                  CALL    SEQMM
002120  005046                          CLR     -(SP)                   ;RESET BR INHIBIT
002122                                  CALL    MRPCR
002126  012767  001000  176422          MOV     #Q$REBK,QR$CR2          ;RE-ARM INTERRUPTS
002134  012767  022000  176422          MOV     #<Q$LDPP+Q$ENOP>,QR$CR2 ;SET LOAD PPS MODE + ENABLE INTERRUPTS
002142  012746  000360                  MOV     #Q$CSEL,-(SP)           ;CLEAR ALL SELECTIONS
002146  052716  001001                  BIS     #<Q$LBD+Q$LBP>,(SP)     ;CLEAR DRIVE AND PULSE
002152  052716  176000                  BIS     #Q$NCLK,(SP)            ;CLEAR ALL NO-CLOCKS
002156  005046                          CLR     -(SP)                   ;SET NOTHING
002160                                  CALL    CSR1
                                ;
                                ;       WAIT FOR INTERRUPT FROM CP
                                ;       USE DMA INTERRUPT (SEE CP MICROCODE FOR QEX)
                                ;
002164                                  WTSE$S  #EFN.3
                                ;
002176                                  CLEF$S  #EFN.3
                                ;
                                ;       RE-ARM INTERRUPTS
                                ;
002210  012767  100400  176422          MOV     #<Q$SM+Q$CHB>,QR$CR2    ;CLEAR INTERRUPT (USE HST BUFFER INT)
002216  012767  101000  176422          MOV     #<Q$SM+Q$REBK>,QR$CR2   ;RE-ARM
002224  012767  160000  176422          MOV     #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2 ;ENABLE
                                ;
```

```
492 002232  005046                1$:     CLR     -(SP)                   ;CLEAR NOTHING IN CSR1
493 002234  012746  176000                MOV     #0$NCLK,-(SP)           ;SET NO-CLOCKS
494 002240                                CALL    CSR1
495 002244  005067  176422                CLR     0R$CR2                  ;SET LOAD MODE
496 002250                                RETURN
```

```
                                        ;
                                        ;
                                        ;       LOAD CP DATA MEMORY.
                                        ;
                                        ;
002252                                  LOADCD:
002252  012767  000003  176424                MOV     #Q$LDCD,QR$LBR          ;MOVE ATTN CODE TO LOD BUS REG
002260  012767  120100  176422                MOV     #<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2 ;SET ATTN CODE READY
002266  016701  176422                1$:     MOV     QR$CR2,R1               ;READ CSR2
002272  032701  000100                        BIT     #Q$ATTN,R1              ;ATTN CLEAR
002276  001373                                BNE     1$                      ;NO, READ AGAIN
                                        ;
002300  016767  175534  176424                MOV     CDADD,QR$LBR            ;CD MEMORY START ADDRESS
002306  012767  120040  176422                MOV     #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
002314  016701  176422                2$:     MOV     QR$CR2,R1               ;READ CSR2
002320  032701  000040                        BIT     #Q$CCCP,R1              ;IS CC TO CP CLEAR
002324  001373                                BNE     2$                      ;NO, READ AGAIN
                                        ;
002326  012767  000001  176424                MOV     #1,QR$LBR               ;TRANSFER COUNT = 1 WORD
002334  012767  120040  176422                MOV     #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
002342  016701  176422                3$:     MOV     QR$CR2,R1               ;READ CSR2
002346  032701  000040                        BIT     #Q$CCCP,R1              ;IS CC TO CP CLEAR
002352  001373                                BNE     3$                      ;NO, READ AGAIN
                                        ;
002354  012767  000044  176424                MOV     #CDDAT,QR$LBR           ;CC MEMORY DATA BUFFER
002362  012767  120040  176422                MOV     #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
                                        ;
                                        ;       WAIT FOR INTERRUPT FROM CP
                                        ;
002370                                  WTSE$S  #EFN.3
                                        ;
002402                                  CLEF$S  #EFN.3
                                        ;
                                        ;       RE-ARM INTERRUPTS
                                        ;
002414  005267  175420                INC     CDADD                   ;BUMP CP DATA MEMORY ADDRESS
002420  012767  100400  176422                MOV     #<Q$SM+Q$CHB>,QR$CR2    ;CLEAR INTERRUPT (USE HIT BUFFER INT)
002426  012767  101000  176422                MOV     #<Q$SM+Q$REBK>,QR$CR2   ;RE-ARM
002434  012767  160000  176422                MOV     #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2 ;ENABLE
002442                                  RETURN
                                        ;
000001                                  .END
```

```
ALUCKE=·040000
ALUOE·=·004000
A01   =·010000
BASE·  =·****** GX·
BINWD·=·****** CX·
BITVAL=·000000
BIT0  =·000001
BIT1  =·000002·
BIT10 =·002000
BIT11 =·004000
BIT12·=·010000
BIT13 =·020000
BIT14 =·040000
BIT15 =·100000
BIT2·  =·000004
BIT3  =·000010
BIT4  =·000020
BIT5  =·000040
BIT6  =·000100
BIT7  =·000200
BIT8·  =·000400
BIT9  =·001000
BUFSET=·****** GX·
BYTE0 =·000000
BYTE1 =·000001
BYTE10=·000012·
BYTE11=·000013·
BYTE12=·000014
BYTE13=·000015
BYTE14=·000016
BYTE15=·000017
BYTE16=·000020
BYTE17=·000021
BYTE18=·000022·
BYTE19=·000023
BYTE2·=·000002·
BYTE20=·000024
BYTE21=·000025
BYTE22=·000026
BYTE23=·000027
BYTE24=·000030
BYTE25=·000031
BYTE26=·000032·
BYTE27=·000033
BYTE28=·000034
BYTE29=·000035
BYTE3 =·000003
BYTE30=·000036
BYTE31=·000037
BYTE32=·000040
BYTE33=·000041
BYTE34=·000042·
BYTE35=·000043
BYTE36=·000044
BYTE37=·000045
BYTE38=·000046
BYTE39=·000047
BYTE4 =·000004
BYTE40=·000050
BYTE41=·000051
BYTE42=·000052·
BYTE43=·000053
BYTE44=·000054
BYTE45=·000055
BYTE46=·000056
BYTE47=·000057
BYTE48=·000060
BYTE49=·000061
BYTE5 =·000005
BYTE50=·000062·
BYTE51=·000063
BYTE52=·000064
BYTE53=·000065
BYTE54=·000066
BYTE55=·000067
BYTE56=·000070
BYTE57=·000071
BYTE58=·000072·
BYTE59=·000073
BYTE6 =·000006
BYTE60=·000074
BYTE61=·000075
BYTE62=·000076
BYTE63=·000077
BYTE64=·000100
BYTE65=·000101
BYTE66=·000102·
BYTE67=·000103
BYTE68=·000104
BYTE69=·000105
BYTE7 =·000007
BYTE70=·000106
BYTE71=·000107
BYTE72=·000110
BYTE73=·000111
BYTE74=·000112·
BYTE75=·000113
BYTE76=·000114
BYTE77=·000115
BYTE78=·000116
BYTE79=·000117
BYTE8 =·000010
BYTE80=·000120
BYTE81=·000121
BYTE82=·000122·
BYTE83=·000123
BYTE84=·000124
BYTE85=·000125
BYTE86=·000126
BYTE87=·000127
BYTE88=·000130
BYTE89=·000131
BYTE9 =·000011
BYTE90=·000132·
BYTE91=·000133
BYTE92=·000134
BYTE93=·000135
BYTE94=·000136
BYTE95=·000137
BYTE96=·000140
BYTE97=·000141
BYTE98=·000142·
BYTE99=·000143
BYTVAL=·000144
CBKALL=·001000
CBKCLK=·000400
CDADD·  000040R·
CDDAT·  000044R·
CNOBRE=·100000
CODE·   000046R·
CPCCEN=·010000
CPCR·  =·****** GX·
CPREAD=·040000
CPWRTE=·020000
CSADRD=·000004
CSEQCI=·100000
CSOE·  =·000040
CSR1   =·****** GX·
CSWRTE=·000100
DATA1  =·****** GX·
DBR.RD=·000001
DB$CPP=·001457
DB$SPT=·000026
DB$TPC=·000023
DISPGS=·100000
DMAAWR=·000005
DMARRD=·000003
DMARWR=·000004
EFN.3  =·****** GX·
ENBR·  =·010000
ERR11  =·****** GX·
ERR4   =·****** GX·
ERR5   =·****** GX·
ERR7   =·****** GX·
FAHIGH=·****** GX·
FAL·    000700R·
FALOW·=·****** GX·
FIND·  =·****** GX·
HANG·  =·****** GX·
INCVAL=·****** GX·
KILL·  =·****** GX·
LBPP·  =·****** GX·
LD4     000050RG·
LD4FC·  000672RG·
LD4FP·  000662RG·
LD4LN·=·000010 G·
LD4QL·  000332RG·
LD4QR·  001164RG·
LD4QW·  000322RG·
LD400   001450RG·
LD401   001456RG·
LD4Q2·  001464RG·
LD4TBL· 000000RG·
LD4X·   001760R·
LHHIGH=·****** GX·
LHLOW·=·****** GX·
LOADCD· 002252R·
LOADQX· 002006R·
LOC.EN=·000100
LOC.WA=·040000
LOC.WR=·100000
LOOP·  =·****** GX·
MAREN1=·000001
MAREN2=·004000
MARLOD=·010000
MAROUT=·000002·
MAR.LO=·002000
MAR.QU=·000040
MBKALL=·001000
MBKCLK=·000400
MEND·  =·****** GX·
MMADRD=·000100
MMLEFT=·000002·
MMOE·  =·000004
MMWRTE=·000010
MNOBRE=·100000
MREN1  =·000001
MREN2·=·020000
MRPCR·=·****** GX·
MSTRT·=·****** GX·
MSTRT2=·****** GX·
MSYN·  =·000040
N·     =·000144
PACK·  =·****** GX·
PDATA·=·****** GX·
PGSEL·  001470R·
PLB·   =·000010
PLC·   =·000020
PLD·   =·000030
PLRWR·=·000200
PLR.EN=·000200
PPCR·  =·****** GX·
PPSXX·=·****** GX·
QEX·    000340R·
QR$CR1=·176420
QR$CR2=·176422·
QR$LBR=·176424
QXCNT·  000042R·
QXHIGH=·****** GX·
QXLOW·=·****** GX·
Q$ATTN=·000100
Q$CLCO·=·000001
Q$CCLP=·000040
Q$CHB·=·000400
Q$CHRL=·000200
Q$CLR·=·000040
Q$CNC·=·030000
Q$CP··=·000060
Q$CPCC=·000010
Q$CP2·=·000260
Q$CSC·=·010000
Q$CSEL=·000360
Q$CSET=·000002
Q$CSP·=·020000
Q$DMA·=·000001
Q$ENBK=·040000
Q$ENOP=·020000
Q$FAL·=·004000
Q$FC  =·000045
Q$FO  =·000044
Q$FP  =·000046
Q$HBF·=·000002
Q$ICP·=·000006
Q$IHB·=·000003
Q$IHRL=·000002
Q$IMRP=·000007
Q$LBD·=·001000
Q$LBDP=·001001
Q$LBP·=·000001
Q$LDCD=·000003
Q$LDMD=·000004
Q$LDPP=·002000
Q$LHP·=·010000
Q$MNC·=·140000
Q$MR  =·000052
Q$MRP·=·000040
Q$MRP2=·000240
Q$MSC·=·040000
Q$MSET=·000004
Q$MSP·=·100000
Q$NCLK=·176000
Q$PP  =·000100
Q$PPSW=·000320
Q$PP2·=·000300
Q$QHLT=·000013
Q$QL  =·000043
Q$QLA·=·000053
Q$QLB·=·000054
Q$QLR·=·000001
Q$QW  =·000042
Q$RDCD=·000005
Q$RDMD=·000006
Q$REBK=·001000
Q$RNC·=·006000
Q$RSC·=·004000
Q$RSET=·000010
Q$SM  =·100000
Q$SP  =·000120
Q$SP2·=·000340
RGO.EN=·000200
RGO.VA=·020000
RP·    =·****** GX·
RTNPT·=·****** GX·
SCAN  =·****** GX·
SELPG·=·****** GX·
```

```
SEQCS=  ****** GX      T$DRD=  000004      WORD11=  000026      WORD42=  000124      WORD72=  000220
SEQMM=  ****** GX      T$EMEM=  010000     WORD12=  000030      WORD43=  000126      WORD73=  000222
SEQ.CI=  000010        T$FSAA=  000000     WORD13=  000032      WORD44=  000130      WORD74=  000224
STQP=  ****** GX       T$FSAB=  000004     WORD14=  000034      WORD45=  000132      WORD75=  000226
S$CLR=  000000         T$FSAC=  000014     WORD15=  000036      WORD46=  000134      WORD76=  000230
S$LA=  000001          T$FSB2=  000010     WORD16=  000040      WORD47=  000136      WORD77=  000232
S$QB=  000005          T$IB=  000026       WORD17=  000042      WORD48=  000140      WORD78=  000234
S$QR=  000006          T$IBAR=  000024     WORD18=  000044      WORD49=  000142      WORD79=  000236
S$QX=  000004          T$IBE=  020000      WORD19=  000046      WORD5=  000012       WORD8=  000240
S$SR=  000007          T$IBF=  040000      WORD2=  000004       WORD50=  000144      WORD80=  000242
S$S1=  000010          T$ICD=  000040      WORD20=  000050      WORD51=  000146      WORD81=  000244
S$S2=  000014          T$MODE=  004000     WORD21=  000052      WORD52=  000150      WORD82=  000246
TD$CTR=  176370        T$OB=  000036       WORD22=  000054      WORD53=  000152      WORD83=  000250
TD$CTW=  176360        T$OBE=  004000      WORD23=  000056      WORD54=  000154      WORD84=  000252
TD$INL=  004000        T$OBF=  010000      WORD24=  000060      WORD55=  000156      WORD85=  000254
TD$MEM=  000270        T$OBRA=  000034     WORD25=  000062      WORD56=  000160      WORD86=  000256
TD$OAR=  176344        T$OBWA=  000032     WORD26=  000064      WORD57=  000162      WORD87=  000260
TD$OTR=  176346        T$OUTA=  100000     WORD27=  000066      WORD58=  000164      WORD88=  000262
TD$QRD=  000274        T$RBDO=  000200     WORD28=  000070      WORD59=  000166      WORD89=  000022
TD$SW=  176376         T$RNB=  000040      WORD29=  000072      WORD6=  000014       WORD9=  000264
TD$TAR=  176372        T$RSET=  040000     WORD3=  000006       WORD60=  000170      WORD90=  000266
TD$TAW=  176362        T$SC=  000022       WORD30=  000074      WORD61=  000172      WORD91=  000270
TD$TDR=  176374        T$SCLK=  020000     WORD31=  000076      WORD62=  000174      WORD92=  000272
TD$TDW=  176364        T$SEG1=  000000     WORD32=  000100      WORD63=  000176      WORD93=  000274
T$AD=  000020          T$SEG2=  000001     WORD33=  000102      WORD64=  000200      WORD94=  000276
T$BA=  000002          T$SEG3=  000002     WORD34=  000104      WORD65=  000202      WORD95=  000300
T$BD=  000010          T$SO=  000001       WORD35=  000106      WORD66=  000204      WORD96=  000302
T$BSO=  100000         T$UBUS=  100000     WORD36=  000110      WORD67=  000206      WORD97=  000304
T$BT=  000020          T$1CLK=  000400     WORD37=  000112      WORD68=  000210      WORD98=  000306
T$BTAR=  000030        T$8BEN=  000020     WORD38=  000114      WORD69=  000212      WORD99=  000310
T$BTD=  002000         UBD.IN=  000020     WORD39=  000116      WORD7=  000016       WRDVAL=  001000
T$CD=  000100          WORD0=  000000      WORD4=  000010       WORD70=  000214      XTREAD=  001000
T$CLK=  002000         WORD1=  000002      WORD40=  000120      WORD71=  000216      XTWRTE=  000400
T$DISK=  000200        WORD10=  000024     WORD41=  000122

. ABS.  000000    000
        000000    001
PPLD    002444    002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3729 WORDS  ( 15 PAGES)
DYNAMIC MEMORY:  4916 WORDS  ( 18 PAGES)
ELAPSED TIME:  00:00:56
PPLD,PPLD/-SP=[20,1]IM,[20,1]PPLD
```

```
                                .TITLE  RPPR
000000                          .PSECT  PPPR
                        ;
                        ;
                        ;       HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
                        ;       PIPELINED PROCESSORS' TEST ROUTINES
                        ;
                        ;       COMMANDS:
                        ;       PR      PRINT PPS MEMORY CONTENTS
                        ;
                        ;       ONCE A COMMAND HAS BBEN EXECUTED (OR AN ERROR ENCOUNTERED)
                        ;       THIS MODULE RETURNS CONTROL TO THE MODULE PPS AT LOCATION
                        ;       'PPSXX'.
                        ;
                        ;
                        ; **********************
                        ;
                        ;       NOTE:
                        ;       THE COMMANDS TO PRINT QEX MEMORIES CANNOT BE USED
                        ;       UNLESS MRP MICROPROGRAM MEMORY AND CP CONTROL STORE
                        ;       HAVE BEEN LOADED VIA THE 'LOADER' PROGRAM. QEX PRINTING
                        ;       MAKES USE OF CP DATA MEMORY - THE PREVIOUS CONTENTS
                        ;       OF THAT MEMORY ARE DESTROYED.
                        ;
                        ; **********************
                        ;
                        ;
                                .MCALL  WTSE$S,CLEF$S
                        ;
                        ;
                        ;       TABLE OF VALID PPS MEMORY MNEMONICS AND ADDRESSES OF
                        ;       ASSOCIATED ROUTINES
                        ;
000000                  PR4TBL::
000000  121  127                .ASCII  /QW/                    ;QEX WINDOW MEMORY
000002  000342'                .WORD   PR4QW
000004  121  114                .ASCII  /QL/                    ;QEX LOCATION MEMORY
000006  000352'                .WORD   PR4QL
000010  106  120                .ASCII  /FP/                    ;FAL POINTER MEMORY
000012  301356'                .WORD   PR4FP
000014  106  103                .ASCII  /FC/                    ;FAL COUNTER MEMORY
000016  001366'                .WORD   PR4FC
000020  106  117                .ASCII  /FO/                    ;FAL OVERFLOW BITS
000022  001376'                .WORD   PR4FO
000024  121  122                .ASCII  /QR/                    ;QLB REFERENCE PAGE
000026  001554'                .WORD   PR4QR
000030  121  060                .ASCII  /Q0/                    ;QLB PAGE 0
000032  001744'                .WORD   PR4Q0
000034  121  061                .ASCII  /Q1/                    ;QLB PAGE 1
000036  001752'                .WORD   PR4Q1
000040  121  062                .ASCII  /Q2/                    ;QLB PAGE 2
000042  001760'                .WORD   PR4Q2
        000044          PR4LN   ==      <.-PR4TBL>
                        ;
                        ;
000044  000000         CDADD:  .WORD   0                       ;ADDR FOR PRINTING FROM CP DATA MEMORY
000046  000000         CDDAT:  .WORD   0                       ;DATA FOR TRANSFER TO CP DATA MEMORY
```

```
 58 000050  000000                  QXCNT:  .WORD   0                       ;NUMBER OF QEX ADDR/DATA PAIRS
 59 000052  000000                  CODE:   .WORD   0                       ;CODE FOR MEMORY SELECTION
 60                                 ;
 61                                 ;
 62                                 ;       PRINT
 63                                 ;       PERFORM THIRD LEVEL PARSING
 64                                 ;       EG. IN THE COMMAND:
 65                                 ;       PP>PR QW 0
 66                                 ;       PARSE THE 'QW'
 67                                 ;
 68                                 ;
 69 000054                          PPPR4::
 70 000054                                  CALL    FIND                    ;LOCATE MEMORY MNEMONIC IN COMMAND LINE
 71 000060  103004                          BCC     1$                      ;OK, CONTINUE
 72 000062                                  CALL    ERR4                    ;'MISSING OPERAND'
 73 000066  000167  002066                  JMP     PR4X                    ;EXIT
 74                                 ;
 75 000072  012700  000044          1$:     MOV     #PR4LN,R0               ;NUMBER OF TABLE ENTRIES
 76 000076  012702  000000'                 MOV     #PR4TBL,R2              ;POINT TO TABLE
 77 000102                                  CALL    SCAN                    ;MATCH AGAINST COMMAND LINE
 78 000106  103004                          BCC     2$                      ;MATCH WAS MADE
 79 000110                                  CALL    ERR7                    ;'INVALID MEMORY MNEMONIC'
 80 000114  000167  002040                  JMP     PR4X
 81                                 ;
 82                                 ;       SAVE POINTER TO ROUTINE ASSOCIATED WITH THE MEMORY MNEMONIC,
 83                                 ;       SCAN THE REMAINDER OF THE COMMAND LINE FOR CONTROL INFORMATION:
 84                                 ;
 85                                 ;       1. START ADDRESS ONLY, PRINT ONE MEMORY LOCATION.
 86                                 ;          PP>PR QW 0
 87                                 ;       2. START ADDRESS, LOOP INDICATOR, LOOP ON THE READING OF
 88                                 ;          THIS ONE MEMORY LOCATION ONLY, PRINT THE CONTENTS ONLY
 89                                 ;          ONCE.
 90                                 ;          PP>LD QW 0 L
 91                                 ;       3. START ADDRESS, END ADDRESS, NO LOOP, PRINT THE CONTENTS
 92                                 ;          OF MEMORY BETWEEN THE START AND END ADDRESSES.
 93                                 ;          PP>PR QW 0 7
 94                                 ;       4. START ADDRESS, END ADDRESS, LOOP INDICATOR, READ THE CONTENTS
 95                                 ;          OF MEMORY BETWEEN THE START AND END ADDRESSES, HOWEVER, PRINT
 96                                 ;          ONLY THE CONTENTS OF MEMORY AT THE START ADDRESS.
 97                                 ;          PP>PR QW 0 7 L
 98                                 ;
 99 000120  010167  000000G         2$:     MOV     R1,RTNPT                ;SAVE POINTER
100 000124                                  CALL    FIND                    ;LOCATE NON-BLANK IN COMMAND LINE
101 000130  103004                          BCC     3$                      ;OK, CONTINUE
102 000132                                  CALL    ERR4                    ;'MISSING OPERAND'
103 000136  000167  002016                  JMP     PR4X                    ;EXIT
104 000142                         3$:     CALL    PACK                    ;CONVERT COMMAND LINE VALUE TO BINARY
105 000146  103004                          BCC     4$                      ;CONVERSION SUCCESSFUL
106 000150                                  CALL    ERR5                    ;'INVALID NUMERIC VALUE'
107 000154  000167  002000                  JMP     PR4X
108                                ;
109 000160  016767  000000G 000000G 4$:    MOV     BINWD,MSTRT             ;SAVE PRINT START ADDRESS
110 000166  016767  000000G 000000G        MOV     BINWD,MSTR2             ;SAVE IT TWICE
111 000174  012767  177777  000000G        MOV     #-1,MEND                ;INIT END ADDRESS
112                                ;
113                                ;       START ADDRESS HAS BEEN FOUND, SCAN FOR END ADDRESS OR
114                                ;       LOOP INDICATOR (CONDITIONS 2, 3 ABOVE), IF THERE IS NOTHING
```

```
                                        ;       FURTHER IN THE COMMAND LINE, CONDITION 1 IS IN EFFECT.
                                        ;
000202                                          CALL    FIND                    ;SCAN COMMAND LINE
000206  103004                                 BCC     5$                      ;SOMETHING THERE
000210  016767  000000G 000000G                MOV     MSTRT,MEND              ;SET END ADDR = START ADDR
000216  000445                                 BR       9$                     ;JUMP TO RTN
                                        ;
000220  122711  000114          5$:     CMPB    #'L,(R1)                        ;LOOP INDICATOR
000224  001011                                 BNE     6$                      ;NO, MUST BE UPPER ADDRESS
000226  052767  000000G 000000G                BIS     #OUT,BASE               ;SET FLAG FOR OUTPUT CONTROL
000234  016767  000000G 000000G                MOV     MSTRT,MEND              ;SET END ADDR = START ADDR
000242                                          CALL    HANG                    ;HOW TO STOP LOOP
000246  000431                                 BR      9$                      ;JUMP TO RTN
                                        ;
000250                          6$:     CALL    PACK                            ;CONVERT UPPER ADDRESS
000254  103004                                 BCC     7$                      ;OK, CONTINUE
000256                                          CALL    ERR5                    ;INVALID NUMERIC
000262  000167  001672                         JMP     PR4X                    ;EXIT
                                        ;
                                        ;       SAVE END ADDRESS (BINARY)
                                        ;       CHECK FOR LOOP INDICATOR AFTER END ADDRESS (CONDITION 4)
                                        ;
000266  016767  000000G 000000G 7$:     MOV     BINWD,MEND                      ;SET UP END ADDRESS
000274                                          CALL    FIND                    ;CHECK FOR LOOP INDICATOR
000300  103414                                 BCS     9$                      ;NO LOOP
000302  122711  000114                         CMPB    #'L,(R1)                ;CORRECT LOOP INDICATOR
000306  001404                                 BEQ     8$                      ;YES
000310                                          CALL    ERR11
000314  000167  001640                         JMP     PR4X                    ;NO
000320  052767  000000G 000000G 8$:     BIS     #OUT,BASE                       ;SET OUTPUT CONTROL
000326                                          CALL    HANG                    ;HOW TO STOP LOOP
                                        ;
000332  016701  000000G         9$:     MOV     RTNPT,R1                        ;POINT TO ROUTINE
000336  000171  000000                   JMP     @(R1)                           ;JUMP TO ROUTINE
                                        ;
                                        ;
                                        ;       PRINT FROM QEX WINDOW MEMORY
                                        ;       PRINT FROM QEX LOCATION MEMORY
                                        ;
                                        ;
000342  012767  000042  177502  PR4QW:: MOV     #Q$QW,CODE                      ;SET MEMORY CODE = WINDOW
000350  000403                                 BR      PQEX
000352  012767  000043  177472  PR4QL:: MOV     #Q$QL,CODE                      ;SET CODE = LOCATION
                                        ;
000360  316746  000000G         PQEX:   MOV     QXHIGH,-(SP)                    ;SUPPLY UPPER MEMORY LIMIT
000364  016746  000000G                 MOV     QXLOW,-(SP)                     ;LOWER LIMIT
000370                                          CALL    BUFSET                  ;PREPARE FOR LOAD
000374  103002                                 BCC     1$                      ;NO ERROR
000376  000167  001556                         JMP     PR4X                    ;ERROR
                                        ;
                                        ;       CALCULATE THE NUMBER OF WORDS TO TRANSFER
                                        ;       MSTR2 = START ADDRESS
                                        ;       MEND  = END ADDRESS
                                        ;
000402  016700  000000G         1$:     MOV     MSTR2,R0                        ;LOAD START ADDRESS
000406  016701  000000G                 MOV     MEND,R1                         ;LOAD END ADDRESS
000412  160001                          SUB     R0,R1                           ;GET NUMBER OF WORDS TO PRINT
```

```
                                        INC     R1                      ;+1 (BEACUSE OF ZERO-REL ADDRESS)
                                        MOV     R1,QXCNT                ;SAVE COUNT
                                ;
                                ;       WRITE THE FLAG ('0' = READ) AND THE TRANSFER COUNT INTO
                                ;       THE FIRST TWO LOCATIONS OF CP DATA MEMORY.
                                ;
                                        CLR     -(SP)                   ;START MICROCODE AT 0
                                        CALL    SEQCS
                                        CLR     -(SP)                   ;RESET BR INHIBIT
                                        CALL    CPCR
                                        MOV     #377,-(SP)              ;SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
                                        CALL    SEQMM
                                        CLR     -(SP)                   ;RESET BR INHIBIT
                                        CALL    MRPCR
                                        MOV     #Q$REBK,QR$CR2          ;RE-ARM INTERRUPTS
                                        MOV     #<Q$SM+Q$ENOP>,QR$CR2   ;SET SEARCH MODE + ENABLE INTERRUPTS
                                        MOV     #Q$CSEL,-(SP)           ;CLEAR ALL SELECTIONS
                                        BIS     #<Q$LBD+Q$LBP>,(SP)     ;CLEAR DRIVE AND PULSE
                                        BIS     #Q$CNC,(SP)             ;CLEAR CP NO-CLOCK
                                        CLR     -(SP)                   ;SET NOTHING
                                        CALL    CSR1
                                ;
                                        CLR     CDADD                   ;WRITE FLAG TO CD ADDR 0
                                        CLR     CDDAT                   ;FLAG FOR 'READ QEX'
                                        CALL    LOADCD
                                        MOV     QXCNT,CDDAT             ;WRITE QEX ADDR/DATA PAIRS COUNT TO CD LOC 1
                                        CALL    LOADCD
                                ;
                                ;       NOW FILL EVERY OTHER LOCATION IN CP DATA MEMORY
                                ;       WITH A QEX ADDRESS (UP TO THE UPPER LIMIT OF
                                ;       QEX MEMORY AS ENTERED IN THE COMMAND LINE).
                                ;
000540  016767  000000G 177300  2$:     MOV     MSTR2,CDDAT             ;PUT QEX ADDRESS IN CP DATA MEMORY
000546                                  CALL    LOADCD                  ;LOADCD INCREMENTS CDADD
000552  005267  177266                  INC     CDADD                   ;INCREMENT AGAIN - PASS 1 ADDRESS
000556  066767  000000G 000000G        ADD     INCVAL,MSTR2            ;ADVANCE QEX ADDRESS
000564  026767  000000G 000000G        CMP     MSTR2,MEND              ;HAS UPPER MEM LIMIT BEEN REACHED
000572  101762                        BLOS    2$                      ;NO, CONTINUE
                                ;
000574  005046                        CLR     -(SP)                   ;CLEAR NOTHING IN CSR1
000576  012746  176000                MOV     #Q$NCLK,-(SP)           ;SET NO-CLOCKS
000602                                  CALL    CSR1
000606  005067  176422                CLR     QR$CR2                  ;SET LOAD MODE
                                ;
                                ;       RUN THE QEX MICROCODE. MICOCODE READS QEX CONTENTS INTO
                                ;       CP DATA MEMORY SLOTS SKIPPED ABOVE. IE. THE DATA IS READ
                                ;       IN FOLLOWING EACH ADDRESS. THE FIRST DATA WORD IS IN CP
                                ;       DATA MEMORY LOCATION 3.
                                ;
000612                          STRTMC:
000612  016746  177234                MOV     CODE,-(SP)              ;SELECT WINDOW OR LOCATION MEMORY
000616                                  CALL    PPCR
000622  012746  001760                MOV     #1760,-(SP)             ;START CP MICROCODE AT X'3F0'
000626                                  CALL    SEQCS
000632  005046                        CLR     -(SP)                   ;RESET BR INHIBIT
000634                                  CALL    CPCR
000640  012746  000377                MOV     #377,-(SP)              ;SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
```

```
000644                                  CALL    SEQMM
000650  005046                          CLR     -(SP)                   ;RESET BR INHIBIT
000652                                  CALL    MRPCR
000656  012767  001000  176422          MOV     #Q$REBK,QR$CR2          ;RE-ARM INTERRUPTS
000664  012767  022000  176422          MOV     #<Q$LDPP+Q$ENOP>,QR$CR2 ;SET LOAD PPS MODE + ENABLE INTERRUPTS
000672  012746  000360                  MOV     #Q$CSEL,-(SP)           ;CLEAR ALL SELECTIONS
000676  052716  001001                  BIS     #<Q$LBD+Q$LBP>,(SP)     ;CLEAR DRIVE AND PULSE
000702  052716  176000                  BIS     #Q$NCLK,(SP)            ;CLEAR ALL NO-CLOCKS
000706  005046                          CLR     -(SP)                   ;SET NOTHING
000710                                  CALL    CSR1
                                ;
                                ;       WAIT FOR INTERRUPT FROM CP
                                ;
000714                                  WT3E$S  #EFN.3
                                ;
000726                                  CLEF$S  #EFN.3
                                ;
000740  005046                          CLR     -(SP)                   ;CLEAR NOTHING IN CSR1
000742  012746  176000                  MOV     #Q$NCLK,-(SP)           ;SET NO-CLOCKS
000746                                  CALL    CSR1
000752  005067  176422                  CLR     QR$CR2                  ;SET LOAD MODE
                                ;
                                ;       PRINT QEX VALUES FROM CP DATA MEMORY
                                ;
000756  005046                          CLR     -(SP)                   ;START MICROCODE AT 0
000760                                  CALL    SEQCS
000764  005046                          CLR     -(SP)                   ;RESET BR INHIBIT
000766                                  CALL    CPCR
000772  012746  000377                  MOV     #377,-(SP)              ;SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
000776                                  CALL    SEQMM
001002  005046                          CLR     -(SP)                   ;RESET BR INHIBIT
001004                                  CALL    MRPCR
001010  012767  001000  176422          MOV     #Q$REBK,QR$CR2          ;RE-ARM INTERRUPTS
001016  012767  120000  176422          MOV     #<Q$SM+Q$ENOP>,QR$CR2   ;SET SEARCH MODE + ENABLE INTERRUPTS
001024  012746  000360                  MOV     #Q$CSEL,-(SP)           ;CLEAR ALL SELECTIONS
001030  052716  001001                  BIS     #<Q$LBD+Q$LBP>,(SP)     ;CLEAR DRIVE AND PULSE
001034  052716  030000                  BIS     #Q$CNC,(SP)             ;CLEAR CP NO-CLOCK
001040  005046                          CLR     -(SP)                   ;SET NOTHING
001042                                  CALL    CSR1
                                ;
001046  016767  000000G 000000G          MOV     MSTRT,MSTR2             ;REFRESH START ADDRESS
001054  012767  000003  176762          MOV     #3,CDADD                ;START PRINTING FROM CP DATA ADDR 3
001062  012767  000005  176424  10$:     MOV     #Q$RDCD,QR$LBR          ;MOVE ATTN CODE TO LOD BUS REG
001070  012767  120100  176422          MOV     #<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2 ;SET ATTN CODE READY
001076  016701  176422          5$:     MOV     QR$CR2,R1               ;READ CSR2
001102  032701  000100                  BIT     #Q$ATTN,R1              ;ATTN CLEAR
001106  001373                          BNE     5$                      ;NO, READ AGAIN
                                ;
001110  016767  176730  176424          MOV     CDADD,QR$LBR            ;CD MEMORY START ADDRESS
001116  012767  120040  176422          MOV     #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
001124  016701  176422          6$:     MOV     QR$CR2,R1               ;READ CSR2
001130  032701  000040                  BIT     #Q$CCCP,R1              ;IS CC TO CP CLEAR
001134  001373                          BNE     6$                      ;NO, READ AGAIN
                                ;
001136  012767  000001  176424          MOV     #1,QR$LBR               ;TRANSFER COUNT = 1 WORD
001144  012767  120040  176422          MOV     #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
001152  016701  176422          7$:     MOV     QR$CR2,R1               ;READ CSR2
```

```
 286 00115E  032701  000040                  BIT     #Q$CCCP,R1              ;IS CC TO CP CLEAR
 287 001162  001373                          BNE     7$                      ;NO, READ AGAIN
 288                                 ;
 289 001164  012767  000000G 176424          MOV     #DATA1,QR$LBR           ;CC MEMORY DATA BUFFER
 290 001172  012767  120040  176422          MOV     #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
 291                                 ;
 292                                 ;       WAIT FOR INTERRUPT FROM CP
 293                                 ;
 294 001200                                  WTSE$S  #EFN.3
 295                                 ;
 296 001212                                  CLEF$S  #EFN.3
 297                                 ;
 298                                 ;       RE-ARM INTERRUPTS
 299                                 ;
 300 001224  012767  100400  176422          MOV     #<Q$SM+Q$CHB>,QR$CR2    ;CLEAR INTERRUPT (USE HIT BUFFER INT)
 301 001232  012767  101000  176422          MOV     #<Q$SM+Q$REBK>,QR$CR2   ;RE-ARM
 302 001240  012767  160000  176422          MOV     #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2 ;ENABLE
 303                                 ;
 304                                 ;       PRINT ONE QEX WORD
 305                                 ;
 306 001246                                  CALL    PRDATA                  ;PRINT IT
 307                                 ;
 308 001252  062767  000002  176564          ADD     #2,CDADD                ;BUMP CP DATA MEMORY ADDRESS
 309 001260  066767  000000G 000000G         ADD     INCVAL,MSTR2            ;ADVANCE ADDRESS
 310 001266  026767  000000G 000000G         CMP     MSTR2,MEND              ;HAS UPPER MEMORY LIMIT BEEN REACHED
 311 001274  101672                          BLOS    10$                     ;NO, CONTINUE
 312 001276  032767  000000G 000000G         BIT     #LOOP,BASE              ;LOOP ON ?
 313 001304  001402                          BEQ     8$                      ;NO, EXIT
 314 001306  000167  177300                  JMP     STRTMC                  ;START MICROCODE FOR QEX READ
 315                                 ;
 316                                 ;       RE-ARM INTERRUPTS
 317                                 ;
 318 001312  012767  100400  176422  8$:     MOV     #<Q$SM+Q$CHB>,QR$CR2    ;CLEAR INTERRUPT (USE HIT BUFFER INT)
 319 001320  012767  101000  176422          MOV     #<Q$SM+Q$REBK>,QR$CR2   ;RE-ARM
 320 001326  012767  160000  176422          MOV     #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2 ;ENABLE
 321                                 ;
 322 001334  005046                          CLR     -(SP)                   ;CLEAR NOTHING IN CSR1
 323 001336  012746  176000                  MOV     #Q$NCLK,-(SP)           ;SET NO-CLOCKS
 324 001342                                  CALL    CSR1
 325 001346  005067  176422                  CLR     QR$CR2                  ;SET LOAD MODE
 326 001352  000167  000602                  JMP     PR4X                    ;EXIT
 327                                 ;
 328                                 ;
 329                                 ;       PRINT FROM FAL POINTER MEMORY
 330                                 ;       PRINT FROM FAL COUNTER MEMORY
 331                                 ;       PRINT FROM FAL OVERFLOW MEMORY
 332                                 ;
 333                                 ;
 334 001356  012767  000046  176466  PR4FP::  MOV     #Q$FP,CODE              ;SET MEMORY CODE = POINTER
 335 001364  000407                          BR      PFAL
 336 001366  012767  000045  176456  PR4FC::  MOV     #Q$FC,CODE              ;SET CODE = COUNTER
 337 001374  000403                          BR      PFAL
 338 001376  012767  000044  176446  PR4FO::  MOV     #Q$FO,CODE              ;SET CODE = OVERFLOW
 339                                 ;
 340 001404  016746  000000G         PFAL:   MOV     FAHIGH,-(SP)            ;SUPPLY MEMORY UPPER LIMIT
 341 001410  316746  000000G                 MOV     FALOW,-(SP)             ;LOWER LIMIT
 342 001414                                  CALL    BUFSET                  ;PREPARE FOR LOAD
```

```
343 001420  103006                          BCC···  1$                  ;OK, CONTINUE·
344 001422· 012746  077777                  MOV·    #077777,-(SP)       ;VALUE·FOR QCL·POINTER·
345 001426                                   CALL·   STQP                ;LOAD·QCL·POINTER·
346 001432· 000167  000522                  JMP·    PR4X
347                                  ;
348 001436  016746  000000G·         1$:     MOV·    MSTR2,-(SP)         ;LOAD·ADDR INTO·QCL·POINTER·
349 001442·                                  CALL·   STQP
350 001446  016746  176400                   MOV·    CODE,-(SP)          ;SELECT·MEMORY·
351 001452·                                  CALL·   PPCR                ;WRITE·SELECTION·TO·CONTROL·REG·
352·001456                                   CALL·   PPLB                ;DO·PPS·TO·LOD·BUS·
353 001462· 012667  000000G·                MOV·    (SP)+,DATA1         ;GET·WORD·FROM·FAL·
354 001466                                   CALL·   PRDATA·             ;PRINT·IT·
355 001472· 012746  000040                   MOV·    #Q$CLR,-(SP)        ;CLEAR·PPS
356 001476                                   CALL·   PPCR
357                                  ;
358 001502· 066767  000000G·000000G·         ADD·    INCVAL,MSTR2·       ;ADVANCE·ADDRESS·
359 001510  026767  000000G·000000G·         CMP·    MSTR2,MEND·         ;HAS·UPPER MEMORY·LIMIT·BEEN·REACHED·
360 001516  101747                           BLOS·   1$                  ;NO, CONTINUE·
361 001520  032767  000000G·000000G·         BIT·    #LOOP,BASE·         ;LOOP·ON·?
362·001526  001006                           BNE·    2$                  ;YES, CONTINUE·
363 001530  012746  077777                   MOV·    #077777,-(SP)       ;VALUE·FOR QCL·POINTER·
364 001534                                   CALL·   STQP                ;LOAD·QCL·POINTER·
365 001540  000167  000414                   JMP·    PR4X                ;EXIT·
366 001544  016767  000000G·000000G· 2$:     MOV·    MSTRT,MSTR2·        ;INIT·START·ADDRESS·
367 001552· 000731                           BR      1$                  ;AND·REPEAT·PRINT·
368                                  ;
369                                  ;
370                                  ;       PRINT·FROM·QLB·REFERENCE·PAGE·
371                                  ;
372·                                 ;
373 001554                           PR4QR::
374 001554  016746  000000G·                MOV·    LHHIGH,-(SP)        ;SUPPLY·MEMORY·UPPER·LIMIT·
375 001560  016746  000000G·                MOV·    LHLOW,-(SP)         ;LOWER·LIMIT·
376 001564                                   CALL·   BUFSET·             ;PREPARE·FOR·LOAD·
377 001570  103012·                          BCC·    1$                  ;OK, CONTINUE·
378 001572· 012746  000013                   MOV·    #Q$QHLT,-(SP)       ;HALT·
379 001576                                   CALL·   PPCR
380 001602· 012746  002000                   MOV·    #2000,-(SP)         ;X'400' =·ILLEGAL·ADDRESS·
381 001606                                   CALL·   LBPP                ;SEND·TO·HP·
382·001612· 000167  000342                   JMP·    PR4X
383                                  ;
384 001616  012746  000053           1$:     MOV·    #Q$QLA,-(SP)        ;SELECT·ADDRESS·FOR·QLB·PAGE·
385 001622·                                  CALL·   PPCR                ;WRITE·SELECTION·TO·CONTROL·REG·
386 001626  016746  000000G·                MOV·    MSTR2,-(SP)         ;LOAD ADDR INTO·QCL·POINTER·
387 001632·                                  CALL·   LBPP
388 001636  012746  000001                   MOV·    #Q$QLR,-(SP)        ;SELECT·QLB·REF·PAGE
389 001642·                                  CALL·   PPCR
390 001646                                   CALL·   PPLB                ;DO·PPS·TO·LOD·BUS·
391 001652· 012667  000000G·                MOV·    (SP)+,DATA1         ;GET·WORD·FROM·FAL·
392·001656                                   CALL·   PRDATA·             ;PRINT·IT·
393                                  ;
394 001662· 066767  000000G·000000G·         ADD·    INCVAL,MSTR2·       ;ADVANCE·ADDRESS·
395 001670  026767  000000G·000000G·         CMP·    MSTR2,MEND·         ;HAS·UPPER MEMORY·LIMIT·BEEN·REACHED·
396 001676  101747                           BLOS·   1$                  ;NO, CONTINUE·
397 001700  032767  000000G·000000G·         BIT·    #LOOP,BASE·         ;LOOP·ON·?
398 001706  001012·                          BNE·    2$                  ;YES, CONTINUE·
399 001710  012746  000013                   MOV·    #Q$QHLT,-(SP)       ;HALT·
```

```
400 001714                                      CALL·   PPCR
401 001720  012746  002000                      MOV·    #2000,-(SP)             ;X'400' = ILLEGAL ADDRESS·
402 001724                                      CALL·   LBPP                    ;SEND TO HP·
403 001730  000167  000224                      JMP·    PR4X                    ;EXIT·
404 001734  016767  000000G 000000G 2$:         MOV·    MSTRT,MSTR2·            ;INIT START ADDRESS·
405 001742  000725                              BR      1$                      ;AND REPEAT PRINT·
406                                     ;
407                                     ;
408                                     ;
409                                     ;       PRINT FROM QLB PAGES·
410                                     ;
411                                     ;
412 001744                              PR4Q0::
413 001744  012746  000000                      MOV·    #0,-(SP)                ;SELECT PAGE 0
414 001750  000405                              BR      PGSEL2·
415 001752·                             PR4Q1::
416 001752  012746  000001                      MOV·    #1,-(SP)                ;SELECT PAGE 1
417 001756  000402·                             BR      PGSEL2·
418 001760                              PR4Q2::
419 001760  012746  000002                      MOV·    #2,-(SP)                ;SELECT PAGE 2
420                                     ;
421 001764                              PGSEL2: CALL·   SELPG·                  ;SELECT A QLB PAGE·
422 001770  016746  000000G                     MOV·    LHHIGH,-(SP)            ;SUPPLY MEMORY UPPER LIMIT·
423 001774  016746  000000G                     MOV·    LHLOW,-(SP)             ;LOWER LIMIT·
424 002000                                      CALL·   BUFSET·                 ;PREPARE FOR LOAD·
425 002004  103012·                             BCC·    1$                      ;OK, CONTINUE·
426 002006  012746  000013                      MOV·    #Q$QHLT,-(SP)           ;HALT·
427 002012·                                     CALL·   PPCR
428 002016  012746  002000                      MOV·    #2000,-(SP)             ;X'400' = ILLEGAL ADDRESS·
429 002022·                                     CALL·   LBPP                    ;SEND TO HP·
430 002026  000167  000126                      JMP·    PR4X
431                                     ;
432 002032  012746  000053              1$:     MOV·    #Q$QLA,-(SP)            ;SELECT ADDRESS FOR QLB PAGE·
433 002036                                      CALL·   PPCR                    ;WRITE SELECTION TO CONTROL REG·
434 002042  016746  000000G                     MOV·    MSTR2,-(SP)             ;LOAD ADDR INTO QCL POINTER·
435 002046                                      CALL·   LBPP
436 002052  012746  000054                      MOV·    #Q$QLB,-(SP)            ;SELECT QLB PAGES·
437 002056                                      CALL·   PPCR
438 002062·                                     CALL·   PPLB                    ;DO PPS TO LOD BUS·
439 002066  012667  000000G                     MOV·    (SP)+,DATA1             ;GET WORD FROM FAL·
440 002072·                                     CALL·   PRDATA·                 ;PRINT IT·
441                                     ;
442 002076  066767  000000G 000000G·            ADD·    INCVAL,MSTR2·           ;ADVANCE ADDRESS·
443 002104  026767  000000G 000000G·            CMP·    MSTR2,MEND·             ;HAS UPPER MEMORY LIMIT BEEN REACHED·
444 002112  101747                              BLOS·   1$                      ;NO, CONTINUE·
445 002114  032767  000000G 000000G·            BIT·    #LOOP,BASE·             ;LOOP ON ?
446 002122  001012·                             BNE·    2$                      ;YES, CONTINUE·
447 002124  012746  000013·                     MOV·    #Q$QHLT,-(SP)           ;HALT·
448 002130                                      CALL·   PPCR
449 002134  012746  002000                      MOV·    #2000,-(SP)             ;X'400' = ILLEGAL ADDRESS·
450 002140                                      CALL·   LBPP                    ;SEND TO HP·
451 002144  000167  000010                      JMP·    PR4X                    ;EXIT·
452 002150  016767  000000G 000000G 2$:         MOV·    MSTRT,MSTR2·            ;INIT START ADDRESS·
453 002156  000725                              BR      1$                      ;AND REPEAT PRINT·
454                                     ;
455                                     ;
456 002160                              PR4X:
```

```
457 002160  012746  000040            MOV     #Q$CLR,-(SP)        ;CLEAR PPS
458 002164                            CALL    PPCR
459 002170  042767  000000C 000000G   BIC     #<ONCE+OUT>,BASE    ;CLEAR PRINT CONTROL FLAGS
460 002176                            CALL    KILL                ;KILL AST (IF THERE WAS ONE)
461 002202  000167  000000G           JMP     PPSXX
```

```
                                        ;
                                        ;
                                        ;       LOAD·CP·DATA MEMORY·
                                        ;
                                        ;
002206                                  LOADCD:
002206  012767  000003  176424                  MOV·    #Q$LDCD,QR$LBR·         ;MOVE·ATTN CODE·TO·LOD·BUS·REG
002214  012767  120100  176422·                 MOV·    #<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2·;SET ATTN CODE·READY
002222· 016701  176422                  1$:     MOV·    QR$CR2,R1               ;READ CSR2
002226· 032701  000100                          BIT·    #Q$ATTN,R1              ;ATTN·CLEAR·
002232· 001373                                  BNE·    1$                      ;NO, READ·AGAIN·
                                        ;
002234  016767  175604  176424                  MOV·    CDADD,QR$LBR·           ;CD·MEMORY START·ADDRESS·
002242· 012767  120040  176422·                 MOV·    #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2·;SET·CC·TO·CP·
002250  016701  176422                  2$:     MOV·    QR$CR2,R1               ;READ CSR2
002254  032701  000040                          BIT·    #Q$CCCP,R1              ;IS·CC·TO·CP·CLEAR·
002260  001373                                  BNE·    2$                      ;NO, READ·AGAIN·
                                        ;
002262· 012767  000001  176424                  MOV·    #1,QR$LBR·              ;TRANSFER·COUNT = 1 WORD·
002270· 012767  120040  176422·                 MOV·    #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2·;SET·CC·TO·CP·
002276  016701  176422                  3$:     MOV·    QR$CR2,R1               ;READ CSR2
002302· 032701  000040                          BIT·    #Q$CCCP,R1              ;IS·CC·TO·CP·CLEAR·
002306  001373                                  BNE·    3$                      ;NO, READ·AGAIN·
                                        ;
002310  012767  000046' 176424                  MOV·    #CDDAT,QR$LBR·          ;CC·MEMORY DATA·BUFFER·
002316  012767  120040  176422·                 MOV·    #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2·;SET·CC·TO·CP·
                                        ;
                                        ;       WAIT·FOR·INTERRUPT·FROM·CP·
                                        ;
002324                                          WTSE$S· #EFN.3
                                        ;
002336                                          CLEF$S· #EFN.3
                                        ;
                                        ;       RE-ARM·INTERRUPTS·
                                        ;
002350  005267  175470                          INC·    CDADD·                  ;BUMP·CP·DATA·MEMORY ADDRESS·
002354  012767  100400  176422·                 MOV·    #<Q$SM+Q$CHB>,QR$CR2·   ;CLEAR·INTERRUPT·(USE·HIT·BUFFER·INT)
002362· 012767  101000  176422·                 MOV·    #<Q$SM+Q$REBK>,QR$CR2·  ;RE-ARM·
002370  012767  160000  176422·                 MOV·    #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2·;ENABLE·
002376                                          RETURN·
                                        ;
        000001                                  .END·
```

```
ALUCKE= 040000          BYTE4  = 000004          BYTE91= 000133          MAROUT= 000002          Q$CLR  = 000040
ALUOE  = 004000          BYTE40= 000050          BYTE92= 000134          MAR.LO= 002000          Q$CNC  = 030000
A01    = 010000          BYTE41= 000051          BYTE93= 000135          MAR.OU= 000040          Q$CP   = 000060
BASE   = ****** GX       BYTE42= 000052          BYTE94= 000136          MBKALL= 001000          Q$CPCC= 000010
BINWD  = ****** GX       BYTE43= 000053          BYTE95= 000137          MBKCLK= 000400          Q$CP2  = 000260
BITVAL= 000000           BYTE44= 000054          BYTE96= 000140          MEND   = ****** GX      Q$CSC  = 010000
BIT0   = 000001          BYTE45= 000055          BYTE97= 000141          MMADRD= 000100          Q$CSEL= 000360
BIT1   = 000002          BYTE46= 000056          BYTE98= 000142          MMLEFT= 000002          Q$CSET= 000002
BIT10  = 002000          BYTE47= 000057          BYTE99= 000143          MMOE   = 000004         Q$CSP  = 020000
BIT11  = 004000          BYTE48= 000060          BYTVAL= 000144          MMWRTE= 000010          Q$DMA  = 000001
BIT12  = 010000          BYTE49= 000061          CBKALL= 001000          MNOBRE= 100000          Q$ENBK= 040000
BIT13  = 020000          BYTE5  = 000005         CBKCLK= 000400          MREN1  = 000001         Q$ENOP= 020000
BIT14  = 040000          BYTE50= 000062          CDADD   000044R     002 MREN2  = 020000         Q$FAL  = 004000
BIT15  = 100000          BYTE51= 000063          CDDAT   000046R     002 MRPCR  = ****** GX      Q$FC   = 000045
BIT2   = 000004          BYTE52= 000064          CNOBRE= 100000          MSTRT  = ****** GX      Q$FO   = 000044
BIT3   = 000010          BYTE53= 000065          CODE    000052R     002 MSTR2  = ****** GX      Q$FP   = 000046
BIT4   = 000020          BYTE54= 000066          CPCCEN= 010000          MSYN   = 000040         Q$HBF  = 000002
BIT5   = 000040          BYTE55= 000067          CPCR   = ****** GX      N      = 000144         Q$ICP  = 000006
BIT6   = 000100          BYTE56= 000070          CPREAD= 040000          ONCE   = ****** GX      Q$IHB  = 000003
BIT7   = 000200          BYTE57= 000071          CPWRTE= 020000          OUT    = ****** GX      Q$IHRL= 000002
BIT8   = 000400          BYTE58= 000072          CSADRD= 000004          PACK   = ****** GX      Q$IMRP= 000007
BIT9   = 001000          BYTE59= 000073          CSEQCI= 100000          PFAL    001404R     002 Q$LBD  = 001000
BUFSET= ****** GX        BYTE6  = 000006         CSOE   = 000040         PGSEL2  001764R     002 Q$LBDP= 001001
BYTE0  = 000000          BYTE60= 000074          CSR1   = ****** GX      PLB    = 000010         Q$LBP  = 000001
BYTE1  = 000001          BYTE61= 000075          CSWRTE= 000100          PLC    = 000020         Q$LDCD= 000003
BYTE10= 000012           BYTE62= 000076          DATA1  = ****** GX      PLD    = 000030         Q$LDMD= 000004
BYTE11= 000013           BYTE63= 000077          DBR.RD= 000001          PLRWR  = 000200         Q$LDPP= 002000
BYTE12= 000014           BYTE64= 000100          DB$CPP= 001457          PLR.EN= 000200          Q$LHP  = 010000
BYTE13= 000015           BYTE65= 000101          DB$SPT= 000026          PPCR   = ****** GX      Q$MNC  = 140000
BYTE14= 000016           BYTE66= 000102          DB$TPC= 000023          PPLB   = ****** GX      Q$MR   = 000052
BYTE15= 000017           BYTE67= 000103          DISPGS= 100000          PPPR4   000054RG    002 Q$MRP  = 000040
BYTE16= 000020           BYTE68= 000104          DMAAWR= 000005          PPSXX  = ****** GX      Q$MRP2= 000240
BYTE17= 000021           BYTE69= 000105          DMARRD= 000003          PQEX    000360R     002 Q$MSC  = 040000
BYTE18= 000022           BYTE7  = 000007         DMARWR= 000004          PRDATA= ****** GX       Q$MSET= 000004
BYTE19= 000023           BYTE70= 000106          EFN.3  = ****** GX      PR4FC   001366RG    002 Q$MSP  = 100000
BYTE2  = 000002          BYTE71= 000107          ENBR   = 010000         PR4FO   001376RG    002 Q$NCLK= 176000
BYTE20= 000024           BYTE72= 000110          ERR11  = ****** GX      PR4FP   001356RG    002 Q$PP   = 000100
BYTE21= 000025           BYTE73= 000111          ERR4   = ****** GX      PR4LN  = 000044 G       Q$PPSW= 000320
BYTE22= 000026           BYTE74= 000112          ERR5   = ****** GX      PR4OL   000352RG    002 Q$PP2  = 000300
BYTE23= 000027           BYTE75= 000113          ERR7   = ****** GX      PR4OR   001554RG    002 Q$QHLT= 000013
BYTE24= 000030           BYTE76= 000114          FAHIGH= ****** GX       PR4OW   000342RG    002 Q$QL   = 000043
BYTE25= 000031           BYTE77= 000115          FALOW  = ****** GX      PR4Q0   001744RG    002 Q$QLA  = 000053
BYTE26= 000032           BYTE78= 000116          FIND   = ****** GX      PR4Q1   001752RG    002 Q$QLB  = 000054
BYTE27= 000033           BYTE79= 000117          HANG   = ****** GX      PR4Q2   001760RG    002 Q$QLR  = 000001
BYTE28= 000034           BYTE8  = 000010         INCVAL= ****** GX       PR4TBL  000000RG    002 Q$QW   = 000042
BYTE29= 000035           BYTE80= 000120          KILL   = ****** GX      PR4X    002160R     002 Q$RDCD= 000005
BYTE3  = 000003          BYTE81= 000121          LBPP   = ****** GX      QR$CR1= 176420          Q$RDMD= 000006
BYTE30= 000036           BYTE82= 000122          LHHIGH= ****** GX       QR$CR2= 176422          Q$REBK= 001000
BYTE31= 000037           BYTE83= 000123          LHLOW  = ****** GX      QR$LBR= 176424          Q$RNC  = 006000
BYTE32= 000040           BYTE84= 000124          LCADCD  002206R     002 QXCNT   000050R     002 Q$RSC  = 004000
BYTE33= 000041           BYTE85= 000125          LOC.EN= 000100          QXHIGH= ****** GX       Q$RSET= 000010
BYTE34= 000042           BYTE86= 000126          LOC.WA= 040000          QXLOW  = ****** GX      Q$SM   = 100000
BYTE35= 000043           BYTE87= 000127          LOC.WB= 100000          Q$ATTN= 000100          Q$SP   = 000120
BYTE36= 000044           BYTE88= 000130          LOOP   = ****** GX      Q$BCL  = 000001         Q$SP2  = 000340
BYTE37= 000045           BYTE89= 000131          MAREN1= 000001          Q$CCCP= 000040          RGO.EN= 000200
BYTE38= 000046           BYTE9  = 000011         MAREN2= 004000          Q$CHB  = 000400         RGO.VA= 020000
BYTE39= 000047           BYTE90= 000132          MARLOD= 010000          Q$CHRL= 000200          RTNPT  = ****** GX
```

```
PPPR·  M[illegible]0·M1110  27-MAR-80 15:24  PAGE·6-2·
SYMBOL·TA[illegible]E·

SCAN·  =·****** GX·        T$CLK·=·002000      WORD1 =·000002·     WORD40=·000120      WORD71=·000216
SELPG·=·****** GX·         T$DISK=·000200      WORD10=·000024      WORD41=·000122·     WORD72=·000220
SEQCS·=·****** GX·         T$DRD·=·000004      WORD11=·000026      WORD42=·000124      WORD73=·000222
SEQMM·=·****** GX·         T$EMEM=·010000      WORD12=·000030      WORD43=·000126      WORD74=·000224
SEQ.CI=·000010             T$FSAA=·000000      WORD13=·000032·     WORD44=·000130      WORD75=·000226
STOP·  =·****** GX·        T$FSAB=·000004      WORD14=·000034      WORD45=·000132·     WORD76=·000230
STRTMC·  000612R·     002·T$FSAC=·000014      WORD15=·000036      WORD46=·000134      WORD77=·000232
S$CLR·=·000000             T$FSB2=·000010      WORD16=·000040      WORD47=·000136      WORD78=·000234
S$LA·  =·000001            T$IB·  =·000026     WORD17=·000042·     WORD48=·000140      WORD79=·000236
S$QB·  =·000005            T$IBAR=·000024      WORD18=·000044      WORD49=·000142·     WORD8 =·000020
S$QR·  =·000006            T$IBE·=·020000      WORD19=·000046      WORD5 =·000012·     WORD80=·000240
S$QX·  =·000004            T$IBF·=·040000      WORD2·=·000004      WORD50=·000144      WORD81=·000242
S$SR·  =·000007            T$ICD·=·000040      WORD20=·000050      WORD51=·000146      WORD82=·000244
S$S1   =·000010            T$MODE=·004000·     WORD21=·000052·     WORD52=·000150      WORD83=·000246
S$S2·  =·000014            T$OB·  =·000036     WORD22=·000054      WORD53=·000152·     WORD84=·000250
TD$CTR=·176370             T$OBE·=·004000      WORD23=·000056      WORD54=·000154      WORD85=·000252
TD$CTW=·176360             T$OBF·=·010000      WORD24=·000060      WORD55=·000156      WORD86=·000254
TD$INL=·004000             T$OBRA=·000034      WORD25=·000062·     WORD56=·000160      WORD87=·000256
TD$MEM=·000270             T$OBWA=·000032·     WORD26=·000064      WORD57=·000162·     WORD88=·000260
TD$OAR=·176344             T$OUTA=·100000      WORD27=·000066      WORD58=·000164      WORD89=·000262
TD$OTR=·176346             T$RBDO=·000200      WORD28=·000070      WORD59=·000166      WORD9 =·000022
TD$QRD=·000274             T$RNB·=·000040      WORD29=·000072·     WORD6 =·000014      WORD90=·000264
TD$SW·=·176376             T$RSET=·040000      WORD3 =·000006      WORD60=·000170      WORD91=·000266
TD$TAR=·176372·            T$SC·  =·000022·    WORD30=·000074      WORD61=·000172·     WORD92=·000270
TD$TAW=·176362·            T$SCLK=·020000      WORD31=·000076      WORD62=·000174      WORD93=·000272
TD$TDR=·176374             T$SEG1=·000000      WORD32=·000100      WORD63=·000176      WORD94=·000274
TD$TDW=·176364             T$SEG2=·000001      WORD33=·000102·     WORD64=·000200      WORD95=·000276
T$AD·  =·000020            T$SEG3=·000002·     WORD34=·000104      WORD65=·000202·     WORD96=·000300
T$BA·  =·000002·           T$SO·  =·000001     WORD35=·000106      WORD66=·000204      WORD97=·000302
T$BD·  =·000010            T$UBUS=·100000      WORD36=·000110      WORD67=·000206      WORD98=·000304
T$BSO·=·100000             T$1CLK=·000400      WORD37=·000112·     WORD68=·000210      WORD99=·000306
T$BT·  =·000020            T$8BEN=·000020      WORD38=·000114      WORD69=·000212·     WRDVAL=·000310
T$BTAR=·000030             UBD.IN=·000020      WORD39=·000116      WORD7 =·000016      XTREAD=·001000
T$BTD·=·002000             WORD0 =·000000      WORD4 =·000010      WORD70=·000214      XTWRTE=·000400
T$CD·  =·000100

. ABS.  000000     000
        000000     001
PPPR·   002400     002·
ERRORS·DETECTED:  0

VIRTUAL·MEMORY·USED:  3651 WORDS·  (·15 PAGES)
DYNAMIC·MEMORY:  4916 WORDS·  ( 18 PAGES)
ELAPSED·TIME:  00:00:56
PPPR,PPPR/-SP=[20,1]IM.[20,1]PPPR·
```

```
 1
 2 000000                               .TITLE  PPREST
 3                                      .PSECT  PPREST
 4                              ;
 5                              ;
 6                              ;       HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
 7                              ;       PIPELINE PROCESSOR TEST ROUTINES
 8                              ;
 9                              ;       COMMANDS:
10                              ;       ST      STORE INTO A PP REGISTER
11                              ;       RE      READ FROM A PP REGISTER
12                              ;       PS      SWITCH QLB PAGES
13                              ;       ER      ERASE MEMORIES
14                              ;
15                              ;       ONCE A COMMAND HAS BBEN EXECUTED (OR AN ERROR ENCOUNTERED)
16                              ;       THIS MODULE RETURNS CONTROL TO THE MODULE PPS AT LOCATION
17                              ;       'PPSXX'.
18                              ;
19                              ;
20                              ;       STORE
21                              ;       TABLE OF VALID REGISTER MNEMONICS AND ASSOCIATED ROUTINE
22                              ;       ADDRESSES
23 000000                       ;
24 000000      103      122     ST4TBL::
25 000002  000170'                     .ASCII  /CR/                    ;PPS CONTROL REGISTER
26 000004      121      120            .WORD   ST4CR
27 000006  000202'                     .ASCII  /QP/                    ;QCL POINTER
28 000010      115      122            .WORD   ST4QP
29 000012  000214'                     .ASCII  /MR/                    ;MASK REGISTER
30 000014      121      122            .WORD   ST4MR
31 000016  000236'                     .ASCII  /QR/                    ;PPS Q-REGISTER
32         000004              ST4LN   ==      <.-ST4TBL>/4            .WORD   ST4QR
33                             ;
34                             ;       READ
35                             ;       TABLE OF VALID REGISTER MNEMONICS AND ASSOCIATED ROUTINE
36                             ;       ADDRESSES
37                             ;
38 000020                      RE4TBL::
39 000020      123      127            .ASCII  /SW/                    ;STATUS WORD
40 000022  000360'                     .WORD   RE4SW
41         000001              RE4LN   ==      <.-RE4TBL>/4
42                             ;
43                             ;       ERASE
44                             ;
45 000024                      ER4TBL::
46 000024      106      103            .ASCII  /FC/                    ;FAL COUNTER MEMORY
47 000026  000600'                     .WORD   ER4FC
48 000030      121      122            .ASCII  /QR/                    ;QLB REF PAGE
49 000032  000744'                     .WORD   ER4QR
50 000034      121      060            .ASCII  /Q0/                    ;QLB PAGE 0
51 000036  001102'                     .WORD   ER4Q0
52 000040      121      061            .ASCII  /Q1/                    ;QLB PAGE 1
53 000042  001106'                     .WORD   ER4Q1
54 000044      121      062            .ASCII  /Q2/                    ;QLB PAGE 2
55 000046  001114'                     .WORD   ER4Q2
56         000005              ER4LN   ==      <.-ER4TBL>/4
57                             ;
```

```
                                        ;
                                        ;       STORE
                                        ;       PERFORM THIRD LEVEL PARSING
                                        ;       EG. IN THE COMMAND:
                                        ;       PP>ST 0 CR
                                        ;       PARSE THE '0'
                                        ;
                                        ;
000050                                  ST4::
000050                                          CALL    FIND                    ;FIND A OPERAND IN COMMAND LINE
000054  103004                                  BCC     1$                      ;OK, CONTINUE
000056                                          CALL    ERR4                    ;NOTHING THERE
000062  000167  000176                          JMP     ST4X                    ;RETURN TO TOP OF LOOP (PROMPT)
000066                                  1$:     CALL    PACK                    ;CONVERT VALUE IN COMMAND LINE TO BINARY
000072  103004                                  BCC     2$
000074                                          CALL    ERR5
000100  000167  000160                          JMP     ST4X
                                        ;
                                        ;       CONTINUE PARSING.
                                        ;       FIND THE REGISTER MNEMONIC IN THE COMMAND LINE
                                        ;
000104                                  2$:     CALL    FIND                    ;LOCATE A NON-BLANK
000110  103004                                  BCC     3$
000112                                          CALL    ERR4
000116  000167  000142                          JMP     ST4X
                                        ;
                                        ;       MATCH THE MNEMONIC FROM THE COMMAND LINE AGAINST THE
                                        ;       TABLE OF VALID MNEMONICS.
                                        ;
000122  012700  000004                  3$:     MOV     #ST4LN,R0               ;NUMBER OF TABLE ENTRIES
000126  012702  000000'                         MOV     #ST4TBL,R2              ;R2 -> TABLE
000132                                          CALL    SCAN                    ;LOOK FOR REG MNEMONIC
000136  103004                                  BCC     4$                      ;OK, CONTINUE
000140                                          CALL    ERR6
000144  000167  000114                          JMP     ST4X
                                        ;
                                        ;       SAVE THE POINTER TO THE ROUTINE ASSOCIATED WITH THE
                                        ;       REGISTER, R1 -> ROUTINE ADDRESS.
                                        ;       CALL ROUTINE TO SCAN COMMAND LINE FOR LOOP INDICATOR.
                                        ;       EG. PP>ST 0 CR L
                                        ;       LOOP FLAG WILL BE SET IF INDICATOR IS PRESENT.
                                        ;       JUMP TO ROUTINE TO LOAD REGISTER.
                                        ;
000150  010167  000000G                 4$:     MOV     R1,RTNPT                ;SAVE POINTER TO RTN
000154                                          CALL    LOOPR                   ;LOOP?
000160  016701  000000G                 ST4IN:  MOV     RTNPT,R1                ;POINT TO ROUTINE
000164  000171  000000                          JMP     @(R1)                   ;EXECUTE ROUTINE
                                        ;
                                        ;
                                        ;       PPS CONTROL REGISTER
                                        ;
000170                                  ST4CR::
000170  016746  000000G                         MOV     BINWD,-(SP)             ;DATA FOR LOD BUS REG
000174                                          CALL    PPCR                    ;WRITE TO CONTROL REG
000200  000423                                  BR      ST4LP                   ;TEST FOR LOOP
                                        ;
                                        ;       OCL POINTER
```

```
                                        ;
000202·                                 ST4QP::
000202·  016746  000000G·                       MOV·    BINWD,-(SP)          ;WORD·FOR·QCL·POINTER·
000206                                          CALL·   STQP                 ;LOAD·QCL·POINTER·
000212·  000416                                 BR      ST4LP·               ;TEST·FOR·LOOP·
                                        ;
                                        ;       LOAD·MASK·REGISTER·
                                        ;
000214                                  ST4MR::
000214  012746  000052                          MOV·    #0$MR,-(SP)          ;MASK·REGISTER·SELECT·FOR·PP·CR·
000220                                          CALL·   PPCR                 ;LOAD·PP·CONTROL·REG
000224  016746  000000G·                        MOV·    BINWD,-(SP)          ;WORD·FOR·MASK·REGISTER·
000230                                          CALL·   LBPSC·               ;SEND·MASK WORD·
000234  000405                                  BR      ST4LP·               ;TEST·FOR·LOOP·
                                        ;
                                        ;       LOAD·PPS·Q-REG·
                                        ;
000236                                  ST4QR::
000236  016746  000000G·                        MOV·    BINWD,-(SP)          ;WORD·FOR·Q-REG·
000242·                                         CALL·   QREG                 ;STORE·IT·
000246  000400                                  BR      ST4LP·               ;TEST·FOR·LOOP·
                                        ;
                                        ;
000250                                  ST4LP:
000250  032767  000000G·000000G·                BIT·    #LOOP,BASE·          ;REPEAT·?·
000256  001402·                                 BEQ·    ST4X                 ;NO, EXIT·
000260  000167  177674                          JMP·    ST4IN·
                                        ;
000264                                  ST4X:
000264                                          CALL·   KILL
000270  000167  000000G·                        JMP·    PPSXX·
```

```
                                ;
                                ;
                                ;       READ
                                ;       PERFORM THIRD LEVEL PARSING
                                ;       EG. IN THE COMMAND:
                                ;       PP>RE CR
                                ;       PARSE THE 'CR'
                                ;
                                ;
000274                          RE4::
000274                                  CALL    FIND                    ;LOCATE A REG MNEMONIC
000300  103004                         BCC     1$
000302                                  CALL    ERR4
000306  000167  000164                  JMP     RE4X
                                ;
                                ;       MATCH THE REGISTER MNEMONIC FROM THE COMMAND LINE AGAINST
                                ;       THE TABLE OF VALID MNEMONICS
                                ;
000312  012700  000001          1$:     MOV     #RE4LN,R0               ;NUMBER OF TABLE ENTRIES
000316  012702  000020'                 MOV     #RE4TBL,R2              ;R2 -> TABLE
000322                                  CALL    SCAN                    ;LOOK FOR REG MNEMONIC
000326  103004                         BCC     2$                      ;OK, CONTINUE
000330                                  CALL    ERR6
000334  000167  000136                  JMP     RE4X
                                ;
                                ;       SAVE THE POINTER TO THE ROUTINE ASSOCIATED WITH THE
                                ;       REGISTER, R1 -> ROUTINE ADDRESS.
                                ;       CALL ROUTINE TO SCAN COMMAND LINE FOR LOOP INDICATOR.
                                ;       EG. PP>RE CR L
                                ;       LOOP FLAG WILL BE SET IF INDICATOR IS PRESENT.
                                ;       JUMP TO ROUTINE TO READ REGISTER.
                                ;
000340  010167  000000G         2$:     MOV     R1,RTNPT                ;SAVE POINTER TO RTN
000344                                  CALL    LOOPR                   ;LOOP?
000350  016701  000000G         RE4IN:  MOV     RTNPT,R1                ;POINT TO ROUTINE
000354  000171  000000                  JMP     @(R1)                   ;EXECUTE ROUTINE
                                ;
                                ;
                                ;       STATUS WORD
                                ;
000360                          RE4SW::
000360  012746  001001                  MOV     #<0$LBD+0$LBP>,-(SP)    ;CLEAR DRIVE AND PULSE
000364  052716  000360                  BIS     #0$CSEL,(SP)            ;CLEAR SELECT BITS
000370  012746  176000                  MOV     #0$NCLK,-(SP)           ;SET NO-CLOCKS
000374  052716  000320                  BIS     #0$PPSW,(SP)            ;READ PPS STATUS WORD
000400                                  CALL    CSR1
000404  016701  176424                  MOV     OR$LBR,R1               ;READ LOD BUS REG
000410  042701  177774                  BIC     #177774,R1              ;CLEAR ALL BITS EXCEPT LOW 3
                                ;
000414  012746  000320                  MOV     #0$PPSW,-(SP)           ;CLEAR PP STATUS REQUEST
000420  005046                         CLR     -(SP)                   ;SET NOTHING
000422                                  CALL    CSR1
000426  000400                         BR      RE4PUT                  ;PRINT AND TEST LOOP FLAG
                                ;
                                ;
000430                          RE4PUT:
000430  032767  000000G 000000G         BIT     #ONCE,BASE              ;PRINTED ONCE ?
```

```
204 000436  001011                          BNE·    1$                      ;YES, SKIP PRINT·
205 000440  052767  000000G·000000G·        BIS·    #ONCE,BASE·             ;SET·FLAG·FOR·PRINTED·ONCE·
206 000446  012705  000000G·                MOV·    #PRINT,R5               ;POINT·TO·PRINT·LINE
207 000452·                                 CALL·   UNPK                    ;CONVERT·VALUE·IN·R1 FOR·PRINTING·
208 000456                                  CALL·   CONSOL·                 ;PRINT·IT·
209                                 ;
210 000462· 032767  000000G·000000G· 1$:    BIT·    #LOOP,BASE·             ;REPEAT·?·
211 000470  001402·                         BEQ·    RE4X                    ;NO, EXIT·
212·000472· 000167  177652                  JMP·    RE4IN·
213                                 ;
214 000476                          RE4X:
215 000476  042767  000000G·000000G·        BIC·    #ONCE,BASE·             ;CLEAR·PRINT·CONTROL FLAG·
216 000504                                  CALL·   KILL                    ;KILL·AST·
217 000510  000167  000000G·                JMP·    PPSXX·
```

```
                                        ;
                                        ;
                                        ;       ERASE.
                                        ;
000514                                  ER4::
000514                                          CALL    FIND                    ;FIND A NON-BLANK IN COMMAND LINE.
000520  103004                                  BCC     1$                      ;OK, CONTINUE.
000522                                          CALL    ERR4                    ;NOTHING THERE.
000526  000167  000524                         JMP     ER4X                    ;RETURN TO TOP OF LOOP (PROMPT)
                                        ;
000532  012700  000005                  1$:     MOV     #ER4LN,R0               ;NUMBER OF TABLE ENTRIES.
000536  012702  000024*                         MOV     #ER4TBL,R2              ;R2 -> TABLE.
000542                                          CALL    SCAN                    ;LOOK FOR MNEMONIC.
000546  103004                                  BCC     2$                      ;OK, CONTINUE.
000550                                          CALL    ERR6
000554  000167  000476                         JMP     ER4X
                                        ;
000560  010167  000000G                 2$:     MOV     R1,RTNPT                ;SAVE POINTER TO RTN
000564                                          CALL    LOOPR                   ;LOOP?
000570  016701  000000G                 ER4IN:  MOV     RTNPT,R1                ;POINT TO ROUTINE
000574  000171  000000                          JMP     @(R1)                   ;EXECUTE ROUTINE.
                                        ;
                                        ;
                                        ;       CLEAR FAL COUNTERS.
                                        ;
                                        ;
000600                                  ER4FC::
000600  012746  001001                          MOV     #<Q$LBD+Q$LBP>,-(SP)    ;CLEAR DRIVE AND PULSE.
000604  052716  000360                          BIS     #Q$CSEL,(SP)            ;CLEAR SELECTION BITS.
000610  012746  176000                          MOV     #Q$NCLK,-(SP)           ;SET NO-CLOCKS.
000614                                          CALL    CSR1
                                        ;
                                        ;       TURN ON PPS CLOCKS.
                                        ;
000620  012746  006000                          MOV     #Q$RNC,-(SP)            ;CLEAR PPS NO-CLOCKS (START PPS)
000624  005046                                  CLR     -(SP)                   ;SET NOTHING.
000626                                          CALL    CSR1
                                        ;
                                        ;       SET PPS CONTROL REGISTER.
                                        ;
                                        ;
000632  012746  000025                          MOV     #025,-(SP)              ;SET PPCR = X'15'
000636                                          CALL    PPCR
                                        ;
                                        ;       SET LOD BUS REGISTER = 0
                                        ;       SET LOD BUS DRIVE, SELECT PPS.
                                        ;
000642  005067  176424                          CLR     QR$LBR
000646  005046                                  CLR     -(SP)                   ;CLEAR NOTHING.
000650  012746  001300                          MOV     #<Q$PP2+Q$LBD>,-(SP)    ;SELECT PPS AND SET DRIVE.
000654                                          CALL    CSR1
                                        ;
                                        ;       SET FAL LOD ON, SET LOD BUS REG, TURN FAL LOD OFF.
                                        ;
000660  012767  004000  176422                  MOV     #Q$FAL,QR$CR2           ;SET FAL LOD
000666  012767  077777  176424                  MOV     #077777,QR$LBR          ;WRITE 'FAL PARK' TO LOD BUS REG
000674  005067  176422                          CLR     QR$CR2                  ;CLEAR
```

```
                                ;
                                ;       WAIT 2 MS FOR FAL TO FINISH
                                ;       ADD = 3.17US
                                ;       DEC = 2.65US
                                ;       BNE = 1.87US
                                ;
000700  012701  000454                  MOV     #300.,R1                ;EXECUTE 300 TIMES
000704  005000                          CLR     R0                      ;ADD NOTHING
000706  060000                  1$:     ADD     R0,R0
000710  005301                          DEC     R1
000712  001375                          BNE     1$
                                ;
                                ;       CLEAR LOD BUS DRIVE, DESELECT PPS
                                ;
000714  012746  001300                  MOV     #<Q$PP2+Q$LBD>,-(SP)    ;CLEAR DRIVE AND DESELECT
000720  012746  176000                  MOV     #Q$NCLK,-(SP)           ;SET NO-CLOCKS
000724                                  CALL    CSR1
                                ;
                                ;       CLEAR PP CR
                                ;
000730  012746  000040                  MOV     #Q$CLR,-(SP)            ;CODE FOR CLEAR
000734                                  CALL    PPCR
000740  000167  000312                  JMP     ER4X
                                ;
                                ;
                                ;       CLEAR QLB REFERENCE PAGE
                                ;
                                ;
000744                          ER4QR::
000744  012746  001001                  MOV     #<Q$LBD+Q$LBP>,-(SP)    ;CLEAR DRIVE AND PULSE
000750  052716  000360                  BIS     #Q$CSEL,(SP)            ;CLEAR SELECTION BITS
000754  012746  176000                  MOV     #Q$NCLK,-(SP)           ;SET NO-CLOCKS
000760                                  CALL    CSR1
                                ;
                                ;       TURN ON PPS CLOCKS
                                ;
000764  012746  006000                  MOV     #Q$RNC,-(SP)            ;CLEAR PPS NO-CLOCKS (START PPS)
000770  005046                          CLR     -(SP)                   ;SET NOTHING
000772                                  CALL    CSR1
                                ;
                                ;       SET PPS CONTROL REGISTER
                                ;
000776  012746  000141                  MOV     #141,-(SP)              ;SET PPCR = X'61'
001002                                  CALL    PPCR
                                ;
                                ;       SET LOD BUS REG = 0
                                ;       SET LOD BUS DRIVE, SELECT PPS
                                ;
001006  005067  176424                  CLR     QR$LBR                  ;SET LBR = 0
001012  005046                          CLR     -(SP)                   ;CLEAR NOTHING
001014  012746  001300                  MOV     #<Q$PP2+Q$LBD>,-(SP)    ;SELECT PPS AND SET DRIVE
001020                                  CALL    CSR1
                                ;
                                ;       SET LHP START, THEN CLEAR
                                ;
001024  012767  010000  176422          MOV     #Q$LHP,QR$CR2           ;START LHP
001032  005067  176422                  CLR     QR$CR2                  ;CLEAR
```

```
                                        ;
                                        ;       WAIT 2 MS FOR LHP TO FINISH
                                        ;       ADD = 3.17US
                                        ;       DEC = 2.65US
                                        ;       BNE = 1.87US
                                        ;
001036  012701  000454                  MOV     #300.,R1                ;EXECUTE 300 TIMES
001042  005000                          CLR     R0                      ;ADD NOTHING
001044  060000                  1$:     ADD     R0,R0
001046  005301                          DEC     R1
001050  001375                          BNE     1$
                                        ;
                                        ;       CLEAR LOD BUS DRIVE, DESELECT PPS
                                        ;
001052  012746  001300                  MOV     #<Q$PP2+Q$LBD>,-(SP)    ;CLEAR DRIVE AND DESELECT
001056  012746  176000                  MOV     #Q$NCLK,-(SP)           ;SET NO-CLOCKS
001062                                  CALL    CSR1
                                        ;
                                        ;       CLEAR PP CR
                                        ;
001066  012746  000040                  MOV     #Q$CLR,-(SP)            ;CODE FOR CLEAR
001072                                  CALL    PPCR
001076  000167  000154                  JMP     ER4X
                                        ;
                                        ;
                                        ;
                                        ;       CLEAR QLB PAGES
                                        ;
                                        ;
001102                          ER400::
001102  005046                          CLR     -(SP)                   ;SELECT PAGE 0
001104  000405                          BR      PGSEL3
001106                          ER401::
001106  012746  000001                  MOV     #1,-(SP)                ;SELECT PAGE 1
001112  000402                          BR      PGSEL3
001114                          ER402::
001114  012746  000002                  MOV     #2,-(SP)                ;SELECT PAGE 2
                                        ;
001120                          PGSEL3: CALL    SELPG                   ;SELECT QLB PAGE
001124  012746  001001                  MOV     #<Q$LBD+Q$LBP>,-(SP)    ;CLEAR DRIVE AND PULSE
001130  052716  000360                  BIS     #Q$CSEL,(SP)            ;CLEAR SELECTION BITS
001134  012746  176000                  MOV     #Q$NCLK,-(SP)           ;SET NO-CLOCKS
001140                                  CALL    CSR1
                                        ;
                                        ;       TURN ON PPS CLOCKS
                                        ;
001144  012746  006000                  MOV     #Q$RNC,-(SP)            ;CLEAR PPS NO-CLOCKS (START PPS)
001150  005046                          CLR     -(SP)                   ;SET NOTHING
001152                                  CALL    CSR1
                                        ;
                                        ;       SET PPS CONTROL REGISTER
                                        ;
001156  012746  000155                  MOV     #155,-(SP)              ;SET PPCR = X'6D'
001162                                  CALL    PPCR
                                        ;
                                        ;       SET LOD BUS DRIVE, SELECT PPS
                                        ;
```

```
*** SEGMENT:  MRLD

R/W MEM  LIMITS: 072364 073557 001174 00636.
DISK BLK LIMITS: 000076 000077 000002 00002.

MEMORY ALLOCATION SYNOPSIS:

SECTION,                                        TITLE,  IDENT,  FILE,
--------                                        -----   -----   ----
. BLK.:(RW,I,LCL,REL,CON) 072364 000000 00000.
MRLD   :(RW,I,LCL,REL,CON) 072364 001174 00636.
                           072364 001174 00636. MRLD          MRLD.OBJ;1
$$ALVC:(RW,D,LCL,REL,CON) 073560 000000 00000.
$$RTS  :(RW,I,GBL,REL,OVR) 071144 000002 00002.


GLOBAL SYMBOLS:

LD1    072374-R  LD1LN  000002    LD1MD  073010-R  LD1MM  072646-R  LD1TBL 072364-R
```

```
*** SEGMENT: MRPR

R/W MEM  LIMITS: 072364 073473 001110 00584.
DISK BLK LIMITS: 000100 000101 000002 00002.

MEMORY ALLOCATION SYNOPSIS:

SECTION                                                TITLE  IDENT  FILE
-------                                                -----  -----  ----
. BLK.:(RW,I,LCL,REL,CON) 072364 000000 00000.
MRPR  :(RW,I,LCL,REL,CON) 072364 001106 00582.
                          072364 001106 00582. MRPR          MRPR.OBJ;1
$$ALVC:(RW,D,LCL,REL,CON) 073472 000000 00000.
$$RTS :(RW,I,GBL,REL,OVR) 071144 000002 00002.

GLOBAL SYMBOLS:

PR1     072374-R  PR1LN  000002    PR1MD  073062-R  PR1MM  072662-R  PR1TBL 072364-R
```

*** SEGMENT: MRREST

R/W MEM LIMITS: 072364 073657 001274 00700.
DISK BLK LIMITS: 000102 000103 000002 00002.

MEMORY ALLOCATION SYNOPSIS:

SECTION				TITLE	IDENT	FILE
. BLK.:	(RW,I,LCL,REL,CON)	072364	000000 00000.			
MRREST:	(RW,I,LCL,REL,CON)	072364	001274 00700.			
		072364	001274 00700.	MRREST		MRREST.OBJ;1
$$ALVC:	(RW,D,LCL,REL,CON)	073660	000000 00000.			
$$RTS :	(RW,I,GBL,REL,OVR)	071144	000002 00002.			

GLOBAL SYMBOLS:

CLI	073546-R	REI	073064-R	REIMP	073176-R	STI	072462-R	STILPG	073032-R	STIPL	072700-R
LOCIP	073556-R	REIDW	073322-R	REISW	073230-R	STIBK	072614-R	STILS	073024-R	STIPR	072722-R
LOCIS	073562-R	REILN	000004	REITBL	072442-R	STICR	072602-R	STILW	073016-R	STIQR	072744-R
LOCIW	073566-R	REIMA	073150-R	RS1	073516-R	STILN	000011	STIMA	072636-R	STITBL	072364-R

```
*** SEGMENT:  MRBUG

R/W MEM  LIMITS: 072364 073353 000770 00504.
DISK BLK LIMITS: 000104 000104 000001 00001.

MEMORY ALLOCATION SYNOPSIS:

SECTION,                                        TITLE,  IDENT,  FILE,
-------                                         -----   -----   ----
. BLK.:(RW,I,LCL,REL,CON) 072364 000000 00000.
MRBUG  :(RW,I,LCL,REL,CON) 072364 000766 00502.
                           072364 000766 00502.  MRBUG          MRBUG.OBJ;1
$$ALVC:(RW,D,LCL,REL,CON) 073352 000000 00000.
$$RTS  :(RW,I,GBL,REL,OVR) 071144 000002 00002.


GLOBAL SYMBOLS:

AT1     072364-R  GO1     072672-R  OF1     072472-R  SS1     072516-R
```

*** SEGMENT: CP ...

R/W MEM LIMITS: 072014 073517 001504 00836.
DISK BLK LIMITS: 000105 000106 000002 00002.

MEMORY ALLOCATION SYNOPSIS:

SECTION				TITLE	IDENT	FILE
. BLK.	(RW,I,LCL,REL,CON)	072014	000000	00000.		
CP	(RW,I,LCL,REL,CON)	072014	001312	00714.		
		072014	001312	00714.	CP	CP.OBJ;1
$$ALVC	(RW,D,LCL,REL,CON)	073326	000170	00120.		
$$RTS	(RW,I,GBL,REL,OVR)	071144	000002	00002.		

GLOBAL SYMBOLS:

AT2	073326-R	CPXX	073224-R	HRL	072544-R	LR2	073336-R	PR2	073506-R	RS2	073446-R
BCL	072112-R	FR2	073406-R	INDEX	072110-R	OF2	073346-R	REREG	073324-R	SS2	073376-R
CL2	073426-R	GO2	073416-R	LDREG	073322-R	PB2	073476-R	RE2	073436-R	ST2	073456-R
CP	073144-R	HLB	072144-R	LD2	073466-R	PC2	073356-R	RG2	073366-R		

*** SEGMENT: CPBUG1

R/W MEM LIMITS: 073520 074373 000654 00428.
DISK BLK LIMITS: 000107 000107 000001 00001.

MEMORY ALLOCATION SYNOPSIS:

SECTION				TITLE	IDENT	FILE
. BLK::	(RW,I,LCL,REL,CON)	073520	000000 00000.			
CPBUG1:	(RW,I,LCL,REL,CON)	073520	000654 00428.			
		073520	000654 00428.	CPBUG1		CPBUG1.OBJ;1
$$ALVC:	(RW,D,LCL,REL,CON)	074374	000000 00000.			
$$RTS::	(RW,I,GBL,REL,OVR)	071144	000002 00002.			

GLOBAL SYMBOLS:

AT2 073524-R LR2 073656-R OF2 073632-R PC2 074010-R RG2 073736-R SS2 074072-R

```
AIDQR.TSK          MEMORY ALLOCATION MAP  TKB           PAGE
CPBUG2                    27-MAR-80

*** SEGMENT: CPBUG2

R/W MEM  LIMITS: 073520 076167 002450 01320.
DISK BLK LIMITS: 000110 000112 000003 00003.

MEMORY ALLOCATION SYNOPSIS:

SECTION                                        TITLE  IDENT  FILE
-------                                        -----  -----  ----
. BLK.:(RW,I,LCL,REL,CON) 073520 000000 00000.
CPBUG2:(RW,I,LCL,REL,CON) 073520 002450 01320.
                          073520 002450 01320. CPBUG2        CPBUG2.OBJ;1
$$ALVC:(RW,D,LCL,REL,CON) 076170 000000 00000.
$$RTS :(RW,I,GBL,REL,OVR) 071144 000002 00002.

GLOBAL SYMBOLS:

FR2    074510-R  GO2    073656-R
```

*** SEGMENT: CPREST

R/W MEM LIMITS: 073520 075327 001610 00904.
DISK BLK LIMITS: 000113 000114 000002 00002.

MEMORY ALLOCATION SYNOPSIS:

SECTION				TITLE	IDENT	FILE
. BLK.:(RW,I,LCL,REL,CON)	073520	000000	00000.			
CPREST:(RW,I,LCL,REL,CON)	073520	001606	00902.			
	073520	001686	00902.	CPREST		CPREST.OBJ;1
$$ALVC:(RW,D,LCL,REL,CON)	075326	000000	00000.			
$$RTS :(RW,I,GBL,REL,OVR)	071144	000002	00002.			

GLOBAL SYMBOLS:

CL2	075316-R	RE2RC	075102-R	RE2R4	075142-R	ST2	074000-R	ST2PB	074302-R	ST2RF	074524-R	ST2R7	074564-R
RE2	074664-R	RE2RD	075076-R	RE2R5	075136-R	ST2BK	074134-R	ST2PC	074334-R	ST2R0	074620-R	ST2R8	074560-R
RE2DD	075034-R	RE2RE	075072-R	RE2R6	075132-R	ST2CR	074120-R	ST2PD	074366-R	ST2R1	074614-R	ST2R9	074554-R
RE2LN	000023	RE2RF	075066-R	RE2R7	075126-R	ST2DA	074420-R	ST2RA	074550-R	ST2R2	074610-R	ST2TBL	073520-R
RE2MA	074750-R	RE2R0	075162-R	RE2R8	075122-R	ST2DD	074462-R	ST2RB	074544-R	ST2R3	074604-R		
RE2MP	075000-R	RE2R1	075156-R	RE2R9	075116-R	ST2LN	000031	ST2RC	074540-R	ST2R4	074600-R		
RE2RA	075112-R	RE2R2	075152-R	RE2TBL	073664-R	ST2MA	074160-R	ST2RD	074534-R	ST2R5	074574-R		
RE2RB	075106-R	RE2R3	075146-R	RS2	075266-R	ST2PA	074250-R	ST2RE	074530-R	ST2R6	074570-R		

*** SEGMENT: CPLD

R/W MEM LIMITS: 073520 075407 001670 00952.
DISK BLK LIMITS: 000115 000116 000002 00002.

MEMORY ALLOCATION SYNOPSIS:

SECTION				TITLE	IDENT	FILE
. BLK:	(RW,I,LCL,REL,CON)	073520	000000 00000.			
CPLD	(RW,I,LCL,REL,CON)	073520	001670 00952.			
		073520	001670 00952.	CPLD		CPLD.OBJ;1
$$ALVC:	(RW,D,LCL,REL,CON)	075410	000000 00000.			
$$RTS	(RW,I,GBL,REL,OVR)	071144	000002 00002.			

GLOBAL SYMBOLS:

LD2 073540-R LD2BL 074634-R LD2CD 074212-R LD2CS 074012-R LD2HL 074454-R LD2LN 000004 LD2TBL 073520-R

*** SEGMENT: CPPR

R/W MEM LIMITS: 073520 075303 001564 00884.
DISK BLK LIMITS: 000117 000120 000002 00002.

MEMORY ALLOCATION SYNOPSIS:

SECTION				TITLE	IDENT	FILE
. BLK:	(RW,I,LCL,REL,CON)	073520	000000	00000.		
CPPR	(RW,I,LCL,REL,CON)	073520	001564	00884.		
		073520	001564	00884.	CPPR	CPPR.OBJ;1
$$ALVC:	(RW,D,LCL,REL,CON)	075304	000000	00000.		
$$RTS	(RW,I,GBL,REL,OVR)	071144	000002	00002.		

GLOBAL SYMBOLS:

PB2	074764-R	PB2HB	075034-R	PB2LN	000003	PR2	073544-R	PR2CS	074032-R	PR2TBL	073520-R
PB2BL	075220-R	PB2HL	075126-R	PB2TBL	073530-R	PR2CD	074352-R	PR2LN	000002		

*** SEGMENT: BCE

R/W MEM LIMITS: 072014 072647 000634 00412.
DISK BLK LIMITS: 000121 000121 000001 00001.

MEMORY ALLOCATION SYNOPSIS:

SECTION				TITLE	IDENT	FILE
. BLK.	(RW,I,LCL,REL,CON)	072014	000000	00000.		
BCE	(RW,I,LCL,REL,CON)	072014	000164	00116.		
		072014	000164	00116.	BCE	BCE.OBJ;1
BCREST	(RW,I,LCL,REL,CON)	072200	000446	00294.		
		072200	000446	00294.	BCREST	BCREST.OBJ;1
$$ALVC	(RW,D,LCL,REL,CON)	072646	000000	00000.		
$$RTS	(RW,I,GBL,REL,OVR)	071144	000002	00002.		

GLOBAL SYMBOLS:

BCE	072030-R	RE3C1	072510-R	RE3LN	000003	ST3	072230-R	ST3LB	072370-R
BCEXX	072110-R	RE3C2	072516-R	RE3TBL	072214-R	ST3C1	072350-R	ST3LN	000003
RE3	072424-R	RE3LB	072524-R	RS3	072616-R	ST3C2	072360-R	ST3TBL	072200-R

```
*** SEGMENT: PPS...

R/W MEM  LIMITS: 072014 072567 000554 00364.
DISK BLK LIMITS: 000122 000122 000001 00001.

MEMORY ALLOCATION SYNOPSIS:

SECTION..                                       TITLE..  IDENT.  FILE..
-------                                         -----    -----   ----
. BLK.:(RW,I,LCL,REL,CON) 072014 000000 00000.
PPS    :(RW,I,LCL,REL,CON) 072014 000464 00308.
                           072014 000464 00308. PPS             PPS.OBJ;1
$$ALVC:(RW,D,LCL,REL,CON) 072500 000070 00056.
$$RTS :(RW,I,GBL,REL,OVR) 071144 000002 00002.


GLOBAL SYMBOLS:

CL4    072520-R  ER4    072530-R  PPPR4  072510-R  PPSXX  072132-R  QREG   072222-R  STQP   072360-R
CODE   072050-R  LD4    072500-R  PPS    072052-R  PS4    072540-R  RE4    072550-R  ST4    072560-R
```

```
*** SEGMENT: PPLD

R/W MEM  LIMITS: 072570 075233 002444 01316.
DISK BLK LIMITS: 000123 000125 000003 00003.

MEMORY ALLOCATION SYNOPSIS:

SECTION                                        TITLE  IDENT  FILE
-------
. BLK.:(RW,I,LCL,REL,CON) 072570 000000 00000.
PPLD  :(RW,I,LCL,REL,CON) 072570 002444 01316.
                          072570 002444 01316. PPLD          PPLD.OBJ;1
$$ALVC:(RW,D,LCL,REL,CON) 075234 000000 00000.
$$RTS :(RW,I,GBL,REL,OVR) 071144 000002 00002.


GLOBAL SYMBOLS:

LD4    072640-R  LD4FP  073452-R  LD4QL  073122-R  LD4QW  073112-R  LD4Q1  074246-R  LD4TBL 072570-R
LD4FC  073462-R  LD4LN  000010    LD4QR  073754-R  LD4Q0  074240-R  LD4Q2  074254-R
```

*** SEGMENT: PPPR

R/W MEM LIMITS: 072570 075167 002400 01280.
DISK BLK LIMITS: 000126 000130 000003 00003.

MEMORY ALLOCATION SYNOPSIS:

SECTION				TITLE	IDENT	FILE
. BLK.	(RW,I,LCL,REL,CON)	072570	000000 00000.			
PPPR	(RW,I,LCL,REL,CON)	072570	002400 01280.			
		072570	002400 01280.	PPPR		PPPR.OBJ;1
$$ALVC	(RW,D,LCL,REL,CON)	075170	000000 00000.			
$$RTS	(RW,I,GBL,REL,OVR)	071144	000002 00002.			

GLOBAL SYMBOLS:

.PPR4	072644-R	PR4FD	074166-R	PR4LN	000044	PR40R	074344-R	PR400	074534-R	PR402	074550-R
PR4FC	074156-R	PR4FP	074146-R	PR40L	073142-R	PR40W	073132-R	PR401	074542-R	PR4TBL	072570-R

*** SEGMENT: PPREST

R/W MEM LIMITS: 072570 074167 001400 00768.
DISK BLK LIMITS: 000131 000132 000002 00002.

MEMORY ALLOCATION SYNOPSIS:

SECTION				TITLE	IDENT	FILE
. BLK.:	(RW,I,LCL,REL,CON)	072570	000000 00000.			
PPREST:	(RW,I,LCL,REL,CON)	072570	001376 00766.			
		072570	001376 00766.	PPREST		PPREST:OBJ;1
$$ALVC:	(RW,D,LCL,REL,CON)	074166	000000 00000.			
$$RTS. :	(RW,I,GBL,REL,OVR)	071144	000002 00002.			

GLOBAL SYMBOLS:

CL4	074156-R	ER4LN	000005	ER4Q1	073676-R	PS4	074072-R	RE4SW	073150-R	ST4CR	072760-R	ST4QP	072772-R
ER4	073304-R	ER4QR	073534-R	ER4Q2	073704-R	RE4	073064-R	RE4TBL	072610-R	ST4LN	000004	ST4QR	073026-R
ER4FC	073370-R	ER4Q0	073672-R	ER4TBL	072614-R	RE4LN	000001	ST4	072640-R	ST4MR	073004-R	ST4TBL	072570-R

*** SEGMENT: SP...

R/W MEM LIMITS: 072014 072573 000560 00368.
DISK BLK LIMITS: 000133 000133 000001 00001.

MEMORY ALLOCATION SYNOPSIS:

SECTION				TITLE	IDENT	FILE
. BLK.	(RW,I,LCL,REL,CON)	072014	000000 00000.			
SP.	(RW,I,LCL,REL,CON)	072014	000172 00122.			
		072014	000172 00122.	SP.		SP;OBJ;1
SPSUB.	(RW,I,LCL,REL,CON)	072206	000324 00212.			
		072206	000324 00212.	SPSUB.		SPSUB.OBJ;1
$$ALVC	(RW,D,LCL,REL,CON)	072532	000040 00032.			
$$RTS.	(RW,I,GBL,REL,OVR)	071144	000002 00002.			

GLOBAL SYMBOLS:

CLS.	072552-R	LBSP.	072206-R	LDS	072532-R	SPCR.	072450-R	SPPRS	072542-R	SPSXX	072116-R
CODE.	072034-R	LBSSC.	072304-R	PSS	072562-R	SPLB.	072402-R	SPS.	072036-R.		

```
*** SEGMENT:  SPLD...

R/W MEM  LIMITS: 072574 073647 001054 00556.
DISK BLK LIMITS: 000134 000135 000002 00002.

MEMORY ALLOCATION SYNOPSIS:

SECTION...                                      TITLE..  IDENT.  FILE..
-------                                         -----    -----   ----
. BLK.:(RW,I,LCL,REL,CON) 072574 000000 00000.
SPLD  :(RW,I,LCL,REL,CON) 072574 001052 00554.
                          072574 001052 00554.  SPLD          SPLD.OBJ;1
$$ALVC:(RW,D,LCL,REL,CON) 073646 000000 00000.
$$RTS :(RW,I,GBL,REL,OVR) 071144 000002 00002.


GLOBAL SYMBOLS:

LD5    072624-R  LD5QB  073106-R  LD5QX  073076-R  LD5S1  073312-R  LD5TBL 072574-R
LD5LN  000006    LD5QR  073116-R  LD5SR  073126-R  LD5S2  073322-R
```

```
*** SEGMENT: SPPR

R/W MEM  LIMITS: 072574 073547 000754 00492.
DISK BLK LIMITS: 000136 000136 000001 00001.

MEMORY ALLOCATION SYNOPSIS:

SECTION                                         TITLE  IDENT  FILE
-------                                         -----  -----  ----
. BLK.:(RW,I,LCL,REL,CON) 072574 000000 00000.
SPPR  :(RW,I,LCL,REL,CON) 072574 000752 00490.
                          072574 000752 00490. SPPR           SPPR.OBJ;1
$$ALVC:(RW,D,LCL,REL,CON) 073546 000000 00000.
$$RTS :(RW,I,GBL,REL,OVR) 071144 000002 00002.


GLOBAL SYMBOLS:

PR5LN  000030    PR5QR  073132-R  PR5SR  073142-R  PR5S2  073274-R  SPPR5  072624-R
PR5QB  073122-R  PR5QX  073112-R  PR5S1  073264-R  PR5TBL 072574-R
```

```
*** SEGMENT:  SPREST

R/W MEM  LIMITS: 072574 072613 000020 00016.
DISK BLK LIMITS: 000137 000137 000001 00001.

MEMORY ALLOCATION SYNOPSIS:

SECTION                                          TITLE.  IDENT.  FILE.
-------                                          -----   -----   ----
. BLK.:(RW,I,LCL,REL,CON) 072574 000000 00000.
SPREST:(RW,I,LCL,REL,CON) 072574 000020 00016.
                          072574 000020 00016.  SPREST          SPREST.OBJ;1
$$ALVC:(RW,D,LCL,REL,CON) 072614 000000 00000.
$$RTS  :(RW,I,GBL,REL,OVR) 071144 000002 00002.


GLOBAL SYMBOLS:

CL5     072604-R  PS5     072574-R


*** TASK BUILDER STATISTICS:

    TOTAL WORK FILE REFERENCES: 70839.
    WORK  FILE  READS: 0.
    WORK  FILE WRITES: 0.
    SIZE OF CORE POOL: 15082. WORDS (58. PAGES)
    SIZE OF WORK FILE: 13568. WORDS (53. PAGES)

    ELAPSED TIME:00:00:54
```

```
 390 001166  005046                            CLR·     -(SP)                  ;CLEAR·NOTHING·
 391 001170  012746  001300                     MOV·     #<Q$PP2+Q$LBD>,-(SP)   ;SELECT·PPS·AND·SET·DRIVE·
 392·001174                                     CALL·    CSR1
 393                                    ;
 394                                    ;       SET·LHP·START, THEN·CLEAR·
 395                                    ;
 396 001200  012767  010000  176422·            MOV·     #Q$LHP,QR$CR2·         ;START·LHP
 397 001206  005067  176422                     CLR·     QR$CR2·                ;CLEAR·
 398                                    ;
 399                                    ;       WAIT·2·MS·FOR·LHP·TO·FINISH·
 400                                    ;       ADD·=·3.17US
 401                                    ;       DEC·=·2.65US
 402·                                   ;       BNE·=·1.87US
 403                                    ;
 404 001212·  012701  000454·                   MOV·     #300.,R1               ;EXECUTE·300 TIMES·
 405 001216  005000                             CLR·     R0                     ;ADD·NOTHING·
 406 001220  060000                     1$:     ADD·     R0,R0
 407 001222·  005301                            DEC·     R1
 408 001224  001375                             BNE·     1$
 409                                    ;
 410                                    ;       CLEAR·LOD·BUS·DRIVE, DESELECT·PPS·
 411                                    ;
 412·001226  012746  001300                     MOV·     #<Q$PP2+Q$LBD>,-(SP)   ;CLEAR·DRIVE·AND·DESELECT·
 413 001232·  012746  176000·                   MOV·     #Q$NCLK,-(SP)          ;SET·NO-CLOCKS·
 414 001236                                     CALL·    CSR1
 415                                    ;
 416                                    ;       CLEAR·PP·CR·
 417                                    ;
 418 001242·  012746  000040                    MOV·     #Q$CLR,-(SP)           ;CODE·FOR·CLEAR·
 419 001246                                     CALL·    PPCR
 420 001252·  000167  000000                    JMP·     ER4X
 421                                    ;
 422·                                   ;
 423 001256                             ER4X:
 424 001256  032767  000000G·000000G·           BIT·     #LOOP,BASE·            ;LOOP·ON·ERASE·
 425 001264  001402·                            BEQ·     1$                     ;NO·
 426 001266  000167  177276                     JMP·     ER4IN·                 ;REPEAT·ERASE·
 427                                    ;
 428·001272·                            1$:     CALL·    KILL                   ;KILL·AST·
 429 001276  000167  000000G·                   JMP·     PPSXX·
```

```
                                        ;
                                        ;
                                        ;       PAGE-SWITCH-
                                        ;
                                        ;
001302-                                 PS4::
001302-
001306  103004                                  CALL-   FIND                    ;FIND-A-NON-BLANK-IN-COMMAND-LINE-
001310                                          BCC-    1$                      ;OK, CONTINUE-
001314  000167  000042-                         CALL-   ERR4                    ;NOTHING-THERE-
                                                JMP-    PS4X                    ;RETURN-TO-TOP-OF-LOOP-(PROMPT)
                                        ;
001320                                  1$:     CALL-   PACK                    ;CONVERT-VALUE-IN-COMMAND-LINE-TO-BINARY
001324  103004                                  BCC-    2$
001326                                          CALL-   ERR5
001332-  000167  000024-                        JMP-    PS4X
                                        ;
001336                                  2$:     CALL-   LOOPR-                  ;TEST-FOR-LOOP-OPTION-
001342-  016746  000000G-              3$:     MOV-    BINWD,-(SP)             ;SELECT-A-OLB-PAGE-
001346                                          CALL-   SELPG-
001352-  032767  000000G-000000G-              BIT-    #LOOP,BASE-             ;REPEAT-?-
001360  001370                                  BNE-    3$                      ;YES-
                                        ;
                                        ;
001362-                                 PS4X:
001362-  000167  000000G-                        JMP-    PPSXX-
                                        ;
                                        ;
                                        ;
                                        ;       CALL-HQR-LOADER-
                                        ;
                                        ;
001366                                  CL4::
001366
001372-  000167  000000G-                        CALL-   CL-                     ;CALL-ROUTINE-IN-'MAIN'
                                                JMP-    PPSXX-
                                        ;
         000001                                 .END-
```

```
ALUCKE=·040000          BYTE40=·000050          BYTE92=·000134          MAR,LO=·002000          Q$LBDP=·001001
ALUOE·=·004000          BYTE41=·000051          BYTE93=·000135          MAR,OU=·000040          Q$LBP·=·000001
A01   =·010000          BYTE42=·000052·         BYTE94=·000136          MBKALL=·001000          Q$LDCD=·000003
BASE·  =·****** GX·     BYTE43=·000053          BYTE95=·000137          MBKCLK=·000400          Q$LDMD=·000004
BINWD·=·****** GX·      BYTE44=·000054          BYTE96=·000140          MMADRD=·000100          Q$LDPP=·002000·
BITVAL=·000000          BYTE45=·000055          BYTE97=·000141          MMLEFT=·000002·         Q$LHP·=·010000
BIT0  =·000001          BYTE46=·000056          BYTE98=·000142·         MMOE·  =·000004         Q$MNC·=·140000
BIT1  =·000002·         BYTE47=·000057          BYTE99=·000143          MMWRTE=·000010          Q$MR  =·000052
BIT10 =·002000          BYTE48=·000060          BYTVAL=·000144          MNOBRE=·100000          Q$MRP·=·000040
BIT11 =·004000          BYTE49=·000061          CBKALL=·001000          MRENi =·000001          Q$MRP2=·000240
BIT12·=·010000          BYTE5 =·000005          CBKCLK=·000400          MREN2·=·020000          Q$MSC·=·040000
BIT13 =·020000          BYTE50=·000062·         CL    =·****** GX·      MSYN·  =·000040         Q$MSET=·000004
BIT14 =·040000          BYTE51=·000063          CL4     001366RG·   002·N·     =·000144         Q$MSP·=·100000
BIT15 =·100000          BYTE52=·000064          CNOBRE=·100000          ONCE·  =·****** GX·     Q$NCLK=·176000
BIT2·  =·000004         BYTE53=·000065          CONSOL=·****** GX·      PACK·  =·****** GX·     Q$PP  =·000100
BIT3  =·000010          BYTE54=·000066          CPCCEN=·010000          PGSEL3  001120R·    002·Q$PPSW=·000320
BIT4  =·000020          BYTE55=·000067          CPREAD=·040000          PLB·   =·000010         Q$PP2·=·000300
BIT5  =·000040          BYTE56=·000070          CPWRTE=·020000          PLC·   =·000020         Q$QHLT=·000013
BIT6  =·000100          BYTE57=·000071          CSADRD=·000004          PLD·   =·000030         Q$QL  =·000043
BIT7  =·000200          BYTE58=·000072·         CSEQCI=·100000          PLRWR·=·000200          Q$QLA·=·000053
BIT8·  =·000400         BYTE59=·000073          CSOE·  =·000040         PLR:EN=·000200          Q$QLB·=·000054
BIT9  =·001000          BYTE6 =·000006          CSR1   =·****** GX·     PPCR·  =·****** GX·     Q$QLR·=·000001
BYTE0 =·000000          BYTE60=·000074          CSWRTE=·000100          PPSXX·=·****** GX·      Q$QW  =·000042
BYTE1 =·000001          BYTE61=·000075          DBR,RD=·000001          PRINT·=·****** GX·      Q$RDCD=·000005
BYTE10=·000012·         BYTE62=·000076          DB$CPP=·001457          PS4     001302RG·   002·Q$RDMD=·000006
BYTE11=·000013          BYTE63=·000077          DB$SPT=·000026          PS4X·   001362R·    002·Q$REBK=·001000
BYTE12=·000014          BYTE64=·000100          DB$TPC=·000023          QREG·  =·****** GX·     Q$RNC·=·006000
BYTE13=·000015          BYTE65=·000101          DISPGS=·100000          QR$CR1=·176420          Q$RSC·=·004000
BYTE14=·000016          BYTE66=·000102·         DMAAWR=·000005          QR$CR2=·176422·         Q$RSET=·000010
BYTE15=·000017          BYTE67=·000103          DMARRD=·000003          QR$LBR=·176424          Q$SM  =·100000
BYTE16=·000020          BYTE68=·000104          DMARWR=·000004          Q$ATTN=·000100          Q$SP  =·000120
BYTE17=·000021          BYTE69=·000105          ENBR·  =·010000         Q$BCL·=·000001          Q$SP2·=·000340
BYTE18=·000022·         BYTE7 =·000007          ERR4   =·****** GX·     Q$CCCP=·000040          RE4     000274RG·   002·
BYTE19=·000023          BYTE70=·000106          ERR5   =·****** GX·     Q$CHB·=·000400          RE4IN·  000350R·    002·
BYTE2·=·000002·         BYTE71=·000107          ERR6   =·****** GX·     Q$CHRL=·000200          RE4LN·=·000001 G·
BYTE20=·000024          BYTE72=·000110          ER4     000514RG·   002·Q$CLR·=·000040          RE4PUT· 000430R·    002·
BYTE21=·000025          BYTE73=·000111          ER4FC·  000600RG·   002·Q$CNC·=·030000          RE4SW·  000360RG·   002·
BYTE22=·000026          BYTE74=·000112·         ER4IN·  000570R·    002·Q$CP·  =·000060         RE4TBL· 000020RG·   002·
BYTE23=·000027          BYTE75=·000113          ER4LN·=·000005 G·       Q$CPCC=·000010          RE4X    000476R·    002·
BYTE24=·000030          BYTE76=·000114          ER4QR·  000744RG·   002·Q$CP2·=·000260          RGQ.EN=·000200
BYTE25=·000031          BYTE77=·000115          ER4Q0   001102RG·   002·Q$CSC·=·010000          RGQ:VA=·020000
BYTE26=·000032·         BYTE78=·000116          ER4Q1   001106RG·   002·Q$CSEL=·000360          RSHPT·=·****** GX·
BYTE27=·000033          BYTE79=·000117          ER4Q2·  001114RG·   002·Q$CSET=·000002·         SCAN  =·****** GX·
BYTE28=·000034          BYTE8 =·000010          ER4TBL· 000024RG·   002·Q$CSP =·020000         SELPG·=·****** GX·
BYTE29=·000035          BYTE80=·000120          ER4X·   001256R·    002·Q$DMA·=·000001          SEQ.CI=·000010
BYTE3 =·000003          BYTE81=·000121          FIND·  =·****** GX·     Q$ENBK=·040000          STOP  =·****** GX·
BYTE30=·000036          BYTE82=·000122·         KILL·  =·****** GX·     Q$ENOP=·020000          ST4     000050RG·   002·
BYTE31=·000037          BYTE83=·000123          LBPSC·=·****** GX·      Q$FAL·=·004000          ST4CR·  000170RG·   002·
BYTE32=·000040          BYTE84=·000124          LOC:EN=·000100          Q$FC·  =·000045         ST4IN·  000160R·    002·
BYTE33=·000041          BYTE85=·000125          LOC:WA=·040000          Q$FO·  =·000044         ST4LN·=·000004 G·
BYTE34=·000042·         BYTE86=·000126          LOC:WB=·100000          Q$FP·  =·000046         ST4LP·  000250R·    002·
BYTE35=·000043          BYTE87=·000127          LOOP·  =·****** GX·     Q$HBF·=·000002·         ST4PR·  000214RG·   002·
BYTE36=·000044          BYTE88=·000130          LOOPR·=·****** GX·      Q$ICP·=·000006          ST4QP·  000202RG·   002·
BYTE37=·000045          BYTE89=·000131          MAREN1=·000001          Q$IHB·=·000003          ST4QR·  000236RG    002·
BYTE38=·000046          BYTE9 =·000011          MAREN2=·004000          Q$IHRL=·000002·         ST4TBL· 000000RG·   002·
BYTE39=·000047          BYTE90=·000132·         MARLOD=·010000          Q$IMRP=·000007          ST4X    000264R·    002·
BYTE4 =·000004          BYTE91=·000133          MAROUT=·000002·         Q$LED·=·001000          S$CLR·=·000000
```

```
PPREST. MACRO M1110  27-MAR-80 15:25  PAGE 8-2
SYMBOL TABLE
```

```
S$LA  = 000001          T$FSAC= 000014          WORD13= 000032          WORD43= 000126          WORD73= 000222
S$QB  = 000005          T$FSB2= 000010          WORD14= 000034          WORD44= 000130          WORD74= 000224
S$QR  = 000006          T$IB  = 000026          WORD15= 000036          WORD45= 000132          WORD75= 000226
S$QX  = 000004          T$IBAR= 000024          WORD16= 000040          WORD46= 000134          WORD76= 000230
S$SR  = 000007          T$IBE = 020000          WORD17= 000042          WORD47= 000136          WORD77= 000232
S$S1  = 000010          T$IBF = 040000          WORD18= 000044          WORD48= 000140          WORD78= 000234
S$S2  = 000014          T$ICD = 000040          WORD19= 000046          WORD49= 000142          WORD79= 000236
TD$CTR= 176370          T$MODE= 004000          WORD2 = 000004          WORD5 = 000012          WORD8 = 000020
TD$CTW= 176360          T$OB  = 000036          WORD20= 000050          WORD50= 000144          WORD80= 000240
TD$INL= 004000          T$OBE = 004000          WORD21= 000052          WORD51= 000146          WORD81= 000242
TD$MEM= 000270          T$OBF = 010000          WORD22= 000054          WORD52= 000150          WORD82= 000244
TD$OAR= 176344          T$OBRA= 000034          WORD23= 000056          WORD53= 000152          WORD83= 000246
TD$OTR= 176346          T$OBWA= 000032          WORD24= 000060          WORD54= 000154          WORD84= 000250
TD$QRD= 000274          T$OUTA= 100000          WORD25= 000062          WORD55= 000156          WORD85= 000252
TD$SW = 176376          T$RBDO= 000200          WORD26= 000064          WORD56= 000160          WORD86= 000254
TD$TAR= 176372          T$RNB = 000040          WORD27= 000066          WORD57= 000162          WORD87= 000256
TD$TAW= 176362          T$RSET= 040000          WORD28= 000070          WORD58= 000164          WORD88= 000260
TD$TDR= 176374          T$SC  = 000022          WORD29= 000072          WORD59= 000166          WORD89= 000262
TD$TDW= 176364          T$SCLK= 020000          WORD3 = 000006          WORD6 = 000014          WORD9 = 000022
T$AD  = 000020          T$SEG1= 000000          WORD30= 000074          WORD60= 000170          WORD90= 000264
T$BA  = 000002          T$SEG2= 000001          WORD31= 000076          WORD61= 000172          WORD91= 000266
T$BD  = 000010          T$SEG3= 000002          WORD32= 000100          WORD62= 000174          WORD92= 000270
T$BSO = 100000          T$SO  = 000001          WORD33= 000102          WORD63= 000176          WORD93= 000272
T$BT  = 000020          T$UBUS= 100000          WORD34= 000104          WORD64= 000200          WORD94= 000274
T$BTAR= 000030          T$1CLK= 000400          WORD35= 000106          WORD65= 000202          WORD95= 000276
T$BTD = 002000          T$8BEN= 000020          WORD36= 000110          WORD66= 000204          WORD96= 000300
T$CD  = 000100          UBD.IN= 000020          WORD37= 000112          WORD67= 000206          WORD97= 000302
T$CLK = 002000          UNPK  = ****** GX       WORD38= 000114          WORD68= 000210          WORD98= 000304
T$DISK= 000200          WORD0 = 000000          WORD39= 000116          WORD69= 000212          WORD99= 000306
T$DRD = 000004          WORD1 = 000002          WORD4 = 000010          WORD7 = 000016          WRDVAL= 000310
T$EMEM= 010000          WORD10= 000024          WORD40= 000120          WORD70= 000214          XTREAD= 001000
T$FSAA= 000000          WORD11= 000026          WORD41= 000122          WORD71= 000216          XTWRTE= 000400
T$FSAB= 000004          WORD12= 000030          WORD42= 000124          WORD72= 000220
```

```
. ABS.  000000     000
        000000     001
PPREST  001376     002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3336 WORDS  (14 PAGES)
DYNAMIC MEMORY:  4916 WORDS  ( 18 PAGES)
ELAPSED TIME:  00:00:49
PPREST,PPREST/-SP=[20,1]IM,[20,1]PPREST
```

```
                                .TITLE  SP
000000                          .PSECT  SP
                        ;
                        ;
                        ;       HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
                        ;       SUBDOCUMENT PROCESSORS
                        ;
                        ;       PARSE THE COMMAND LINE AND TRANSFER CONTROL TO ONE OF THE
                        ;       SP SUB-MODULES OR BACK TO QMAIN. CONTROL COULD HAVE BEEN
                        ;       PASSED TO SP IN ONE OF THREE WAYS:
                        ;       1. FROM QMAIN IF SP WAS SELECTED AS THE FIRST PROCESSOR
                        ;          UPON ENTRY TO THE PROGRAM
                        ;          >SP LD QX 0
                        ;       2. FROM QMAIN IF SP WAS SELECTED FROM ONE OF QMAIN'S
                        ;          OTHER SUB-MODULES
                        ;          CP>SP LD QX 0
                        ;       3. ON RETURN FROM ONE OF SP'S SUB-MODULES
                        ;          SP>LD QX 0
                        ;
                        ;       SP PARSES AT THE SECOND LEVEL OF CONTROL (SEE NOTES AT
                        ;       QMAIN ON LEVELS OF CONTROL). IN THE EXAMPLES ABOVE, SP
                        ;       WOULD PARSE 'LD' AND TRANSFER CONTROL TO THE SP SUB-
                        ;       MODULE SPLD. IF SP SHOULD ENCOUNTER A STRING WHICH IS
                        ;       NOT A VALID COMMAND MNEMONIC, SP CONSIDERS THE STRING TO
                        ;       BE A PROCESSOR MNEMONIC AND RETURNS CONTROL TO QMAIN. EG:
                        ;       SP>CP LD CS 0
                        ;       THE STRING 'CP' IS NOT A VALID SP (SECOND LEVEL) COMMAND.
                        ;       SP RETURNS CONTROL TO QMAIN WHICH IN TURN WILL TRANSFER
                        ;       CONTROL TO ITS SUB-MODULE CP.
                        ;
                        ;       SP SUB-MODULES:
                        ;       SPLD    LOAD MEMORIES
                        ;       SPPR    PRINT MEMORY CONTENTS
                        ;       SPREST  ALL OTHER SP COMMANDS
                        ;
                        ;
                                .MCALL  WTSE$S,CLEF$S
```

```
                                        ;
                                        ;
                                        ;       TABLE·OF·SP·COMMAND·MNEMONICS·AND·THEIR·ASSOCIATED·
                                        ;       ROUTINE·ADDRESSES·
                                        ;
                                        ;
000000                                  SPTBL:
000000      114     104                         .ASCII·  /LD/                   ;LOAD·MEMORY·
000002·  000000G·                               .WORD·   LD5
000004      120     122·                        .ASCII·  /PR/                   ;PRINT·FROM·MEMORY·
000006·  000000G·                               .WORD·   SPPR5
000010      120     123                         .ASCII·  /PS/                   ;PAGE·SWITCH·
000012·  000000G·                               .WORD·   PS5
000014      103     114                         .ASCII·  /CL/                   ;CALL·OR·LOADER·
000016·  000000G·                               .WORD·   CL5
            000004              SPNUM·  =·      <.-SPTBL>/4
                                        ;
                                        ;
                                        ;       CODE·FOR·MEMORY·SELECTION·
                                        ;
000020·  000000                 CODE::  .WORD·   0
```

```
                                ;
                                ;
                                ;       SP ROUTINES
                                ;
                                ;
000022                          SPS::
000022                                  CALL    FIND                    ;LOCATE THE COMMAND IN THE COMMAND LINE
000026  103003                          BCC     1$                      ;OK, CONTINUE
000030                                  CALL    ERR3
000034  000422                          BR      SPSXX
000036  022700  000002          1$:     CMP     #2,R0                   ;COMMANDS ARE 2 CHARS
000042  001403                          BEQ     2$
000044                                  CALL    ERR8
000050  000414                          BR      SPSXX                   ;TRY AGAIN
                                ;
000052  012700  000004          2$:     MOV     #SPNUM,R0               ;R0 = NUMBER OF COMMANDS
000056  012702  000000'                 MOV     #SPTBL,R2               ;R2 -> TABLE OF COMMAND MNEMONICS
000062                                  CALL    SCAN                    ;FIND MATCH IN TABLE
000066  103003                          BCC     3$                      ;OK, CONTINUE
000070                                  CALL    ERR12                   ;COMMAND NOT IN TABLE
000074  000402                          BR      SPSXX                   ;TRY AGAIN
                                ;
                                ;       JUMP TO THE ROUTINE THAT GOVERNS THE COMMAND
                                ;
000076  000171  000000          3$:     JMP     @(R1)
                                ;
                                ;
                                ;       LOCAL SP LOOP, INCLUDE PROMPT FOR SP COMMAND.
                                ;
                                ;
000102                          SPSXX::
000102  012767  050123  000000C          MOV     #"SP,GCMBLK+G.DPRM+2    ;MOVE SP NAME TO GCM BLOCK
000110                                  CALL    GCONLY                  ;PROMPT
000114                                  CALL    FIND                    ;LOCATE THE COMMAND IN THE COMMAND LINE
000120  103003                          BCC     1$                      ;OK, CONTINUE
000122                                  CALL    ERR3
000126  000765                          BR      SPSXX
000130  022700  000002          1$:     CMP     #2,R0                   ;COMMANDS ARE 2 CHARS
000134  001403                          BEQ     2$
000136                                  CALL    ERR8
000142  000757                          BR      SPSXX                   ;TRY AGAIN
                                ;
000144  012700  000004          2$:     MOV     #SPNUM,R0               ;R0 = NUMBER OF COMMANDS
000150  012702  000000'                 MOV     #SPTBL,R2               ;R2 -> TABLE OF COMMAND MNEMONICS
000154                                  CALL    SCAN                    ;FIND MATCH IN TABLE
000160  103002                          BCC     3$                      ;OK, CONTINUE
000162  000167  000000G                 JMP     COMXX                   ;RETURN TO 'MAIN', LOOK FOR PROCESSOR MNEMONIC
                                ;
                                ;       JUMP TO THE ROUTINE THAT GOVERNS THE COMMAND
                                ;
000166  000171  000000          3$:     JMP     @(R1)
```

```
114                                     ;
115         000001                              .END·
```

SYMBOL TABLE


```
ALUCKE= 040000
ALUOE = 004000
A01   = 010000
BITVAL= 000000
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BYTE0 = 000000
BYTE1 = 000001
BYTE10= 000012
BYTE11= 000013
BYTE12= 000014
BYTE13= 000015
BYTE14= 000016
BYTE15= 000017
BYTE16= 000020
BYTE17= 000021
BYTE18= 000022
BYTE19= 000023
BYTE2 = 000002
BYTE20= 000024
BYTE21= 000025
BYTE22= 000026
BYTE23= 000027
BYTE24= 000030
BYTE25= 000031
BYTE26= 000032
BYTE27= 000033
BYTE28= 000034
BYTE29= 000035
BYTE3 = 000003
BYTE30= 000036
BYTE31= 000037
BYTE32= 000040
BYTE33= 000041
BYTE34= 000042
BYTE35= 000043
BYTE36= 000044
BYTE37= 000045
BYTE38= 000046
BYTE39= 000047
BYTE4 = 000004
BYTE40= 000050
BYTE41= 000051
BYTE42= 000052
BYTE43= 000053
BYTE44= 000054
BYTE45= 000055
BYTE46= 000056
BYTE47= 000057
BYTE48= 000060
BYTE49= 000061
BYTE5 = 000005
BYTE50= 000062
BYTE51= 000063
BYTE52= 000064
BYTE53= 000065
BYTE54= 000066
BYTE55= 000067
BYTE56= 000070
BYTE57= 000071
BYTE58= 000072
BYTE59= 000073
BYTE6 = 000006
BYTE60= 000074
BYTE61= 000075
BYTE62= 000076
BYTE63= 000077
BYTE64= 000100
BYTE65= 000101
BYTE66= 000102
BYTE67= 000103
BYTE68= 000104
BYTE69= 000105
BYTE7 = 000007
BYTE70= 000106
BYTE71= 000107
BYTE72= 000110
BYTE73= 000111
BYTE74= 000112
BYTE75= 000113
BYTE76= 000114
BYTE77= 000115
BYTE78= 000116
BYTE79= 000117
BYTE8 = 000010
BYTE80= 000120
BYTE81= 000121
BYTE82= 000122
BYTE83= 000123
BYTE84= 000124
BYTE85= 000125
BYTE86= 000126
BYTE87= 000127
BYTE88= 000130
BYTE89= 000131
BYTE9 = 000011
BYTE90= 000132
BYTE91= 000133
BYTE92= 000134
BYTE93= 000135
BYTE94= 000136
BYTE95= 000137
BYTE96= 000140
BYTE97= 000141
BYTE98= 000142
BYTE99= 000143
BYTVAL= 000144
CBKALL= 001000
CBKCLK= 000400
CL5   = ****** GX
CNOBRE= 100000
CODE    000020RG    002
COMXX = ****** GX
CPCCEN= 010000
CPREAD= 040000
CPWRTE= 020000
CSADRD= 000004
CSEQCI= 100000
CSOE  = 000040
CSWRTE= 000100
DBR.RD= 000001
DB$CPP= 001457
DB$SPT= 000026
DB$TPC= 000023
DISPGS= 100000
DMAAWR= 000005
DMARRD= 000003
DMARWR= 000004
ENBR  = 010000
ERR12 = ****** GX
ERR3  = ****** GX
ERR8  = ****** GX
FIND  = ****** GX
GCMBLK= ****** GX
GCONLY= ****** GX
G.DPRM= ****** GX
LD5   = ****** GX
LOC.EN= 000100
LOC.WA= 040000
LOC.WB= 100000
MAREN1= 000001
MAREN2= 004000
MARLOD= 010000
MAROUT= 000002
MAR.LO= 002000
MAR.OU= 000040
MBKALL= 001000
MBKCLK= 000400
MMADRD= 000100
MMLEFT= 000002
MMOE  = 000004
MMWRTE= 000010
MNOBRE= 100000
MREN1 = 000001
MREN2 = 020000
MSYN  = 000040
N     = 000144
PLB   = 000010
PLC   = 000020
PLD   = 000030
PLRWR = 000200
PLR.EN= 000200
PS5   = ****** GX
QR$CR1= 176420
QR$CR2= 176422
QR$LBR= 176424
Q$ATTN= 000100
Q$BCL = 000001
Q$CCCP= 000040
Q$CHB = 000400
Q$CHRL= 000200
Q$CLR = 000040
Q$CNC = 030000
Q$CP  = 000060
Q$CPCC= 000010
Q$CP2 = 000260
Q$CSC = 010000
Q$CSEL= 000360
Q$CSET= 000002
Q$CSP = 020000
Q$DMA = 000001
Q$ENBK= 040000
Q$ENOP= 020000
Q$FAL = 004000
Q$FC  = 000045
Q$FO  = 000044
Q$FP  = 000046
Q$HBF = 000002
Q$ICP = 000006
Q$IHB = 000003
Q$IHRL= 000002
Q$IMRP= 000007
Q$LBD = 001000
Q$LBDP= 001001
Q$LBP = 000001
Q$LDCD= 000003
Q$LDMD= 000004
Q$LDPP= 002000
Q$LHP = 010000
Q$MNC = 140000
Q$MR  = 000052
Q$MRP = 000040
Q$MRP2= 000240
Q$MSC = 040000
Q$MSET= 000004
Q$MSP = 100000
Q$NCLK= 176000
Q$PP  = 000100
Q$PPSW= 000320
Q$PP2 = 000300
[illegible]
Q$QL  = 000043
Q$QLA = 000053
Q$QLB = 000054
Q$QLR = 000001
Q$QW  = 000042
Q$RDCD= 000005
Q$RDMD= 000006
Q$REBK= 001000
Q$RNC = 006000
Q$RSC = 004000
Q$RSET= 000010
Q$SM  = 100000
Q$SP  = 000120
Q$SP2 = 000340
RGQ.EN= 000200
RGQ.VA= 020000
SCAN  = ****** GX
SEQ.CI= 000010
SPNUM = 000004
SPPR5 = ****** GX
SPS     000022RG    002
SPSXX   000102RG    002
SPTBL   000000R     002
S$CLR = 000000
S$LA  = 000001
S$QB  = 000005
S$QR  = 000006
S$QX  = 000004
S$SR  = 000007
S$S1  = 000010
S$S2  = 000014
TD$CTR= 176370
TD$CTW= 176360
TD$INL= 004000
TD$MEM= 000270
TD$OAR= 176344
TD$OTR= 176346
TD$QRD= 000274
TD$SW = 176376
TD$TAR= 176372
TD$TAW= 176362
TD$TDR= 176374
TD$TDW= 176364
T$AD  = 000020
T$BA  = 000002
T$BD  = 000010
T$BSO = 100000
T$BT  = 000020
T$BTAR= 000030
T$BTD = 002000
T$CD  = 000100
T$CLK = 002000
T$DISK= 000200
T$DRD = 000004
T$EMEM= 010000
T$FSAA= 000000
T$FSAB= 000004
T$FSAC= 000014
T$FSB2= 000010
T$IB  = 000026
```

```
SP·····M···0·M1110  27-MAR-80 15:31  PAGE 8-2
SYMBOL·TA··E·

T$IBAR=·000024          WORD10=·000024          WORD34=·000104          WORD57=·000162·         WORD8 =·000020
T$IBE·=·020000          WORDi1=·000026          WORD35=·000106          WORD58=·000164          WORD80=·000240
T$IBF·=·040000          WORD12=·000030          WORD36=·000110          WORD59=·000166          WORD81=·000242
T$ICD·=·000040          WORDi3=·000032·         WORD37=·000112·         WORD6 =·000014          WORD82=·000244
T$MODE=·004000          WORD14=·000034          WORD38=·000114          WORD60=·000170          WORD83=·000246
T$OB··=·00003E          WORD15=·000036          WORD39=·000116          WORD61=·000172·         WORD84=·000250
T$OBE·=·004000          WORD16=·000040          WORD4 =·000010          WORD62=·000174          WORD85=·000252
T$OBF·=·010000          WORD17=·000042·         WORD40=·000120          WORD63=·000176          WORD86=·000254
T$OBRA=·000034          WORD18=·000044          WORD41=·000122·         WORD64=·000200          WORD87=·000256
T$OBWA=·000032·         WORD19=·000046          WORD42=·000124          WORD65=·000202·         WORD88=·000260
T$OUTA=·100000          WORD2·=·000004          WORD43=·000126          WORD66=·000204          WORD89=·000262
T$RBDO=·000200          WORD20=·000050          WORD44=·000130          WORD67=·000206          WORD9 =·000022
T$RNB·=·000040          WORD21=·000052·         WORD45=·000132·         WORD68=·000210          WORD90=·000264
T$RSET=·040000          WORD22=·000054          WORD46=·000134          WORD69=·000212·         WORD91=·000266
T$SC··=·000022·         WORD23=·000056          WORD47=·000136          WORD7 =·000016          WORD92=·000270
T$SCLK=·020000          WORD24=·000060          WORD48=·000140          WORD70=·000214          WORD93=·000272
T$SEG1=·000000          WORD25=·000062·         WORD49=·000142·         WORD71=·000216          WORD94=·000274
T$SEG2=·000001          WORD26=·000064          WORD5 =·000012·         WORD72=·000220          WORD95=·000276
T$SEG3=·000002·         WORD27=·000066          WORD50=·000144          WORD73=·000222·         WORD96=·000300
T$SO··=·000001          WORD28=·000070          WORD51=·000146          WORD74=·000224          WORD97=·000302
T$UBUS=·100000          WORD29=·000072·         WORD52=·000150          WORD75=·000226          WORD98=·000304
T$1CLK=·000400          WORD3 =·000006          WORD53=·000152·         WORD76=·000230          WORD99=·000306
T$8BEN=·000020          WORD30=·000074          WORD54=·000154          WORD77=·000232·         WRDVAL=·000310
UBD.IN=·000020          WORD31=·000076          WORD55=·000156          WORD78=·000234          XTREAD=·001000
WORD0 =·000000          WORD32=·000100          WORD56=·000160          WORD79=·000236          XTWRTE=·000400
WORD1 =·000002·         WORD33=·000102·

. ABS.  000000      000
        000000      001
SP·     000172·     002·
ERRORS·DETECTED:  0

VIRTUAL·MEMORY·USED:  3210 WORDS·  (·13 PAGES)
DYNAMIC·MEMORY:  3860 WORDS·  ( 14 PAGES)
ELAPSED·TIME:  00:00:41
SP›SP/-SP=[20,1]IM,[20,1]SP·
```

```
                                        .TITLE  $PLD
000000                                  .PSECT  SPLD
                                ;
                                ;
                                ;       HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
                                ;       SUBDOCUMENT PROCESSOR TEST ROUTINES
                                ;
                                ;       COMMANDS:
                                ;       LD      LOAD SP MEMORIES
                                ;
                                ;       ONCE A COMMAND HAS BBEN EXECUTED (OR AN ERROR ENCOUNTERED)
                                ;       THIS MODULE RETURNS CONTROL TO THE MODULE SP AT LOCATION
                                ;       'SPXX'.
                                ;
                                ;
                                ;
                                        .MCALL  WTSE$S,CLEF$S
                                ;
                                ;
                                ;       TABLE OF VALID SP MEMORY MNEMONICS AND ADDRESSES OF THEIR
                                ;       ASSOCIATED ROUTINES
                                ;
000000                          LD5TBL::
000000  121  130                        .ASCII  /QX/                    ;QEX MEMORY
000002  000302'                    .WORD   LD5QX
000004  123  122                    .ASCII  /SR/                    ;SUBREAD MEMORY
000006  000332'                    .WORD   LD5SR
000010  123  106                    .ASCII  /SF/                    ;SDLB REFERENCE PAGE
000012  000322'                    .WORD   LD5QR
000014  123  060                    .ASCII  /S0/                    ;SDLB PAGE
000016  000312'                    .WORD   LD5QB
000020  123  061                    .ASCII  /S1/                    ;SIDMEM 1
000022  000516'                    .WORD   LD5S1
000024  123  062                    .ASCII  /S2/                    ;SIDMEM 2
000026  000526'                    .WORD   LD5S2
        000006                  LD5LN   ==      <.-LD5TBL>/4
                                ;
                                ;
                                ;       LOAD
                                ;       PERFORM THIRD LEVEL PARSING
                                ;       EG. IN THE COMMAND:
                                ;       SP>LD QX 0
                                ;       PARSE THE 'QX'
                                ;
                                ;
000030                          LD5::
000030
000034  103004                     CALL    FIND                    ;LOCATE MEMORY MNEMONIC IN COMMAND LINE
000036                              BCC     1$                      ;OK, CONTINUE
000042  000167  000632              CALL    ERR4                    ;'MISSING OPERAND'
                                    JMP     LD5X                    ;EXIT
                                ;
                                ;       MATCH THE MNEMONIC IN THE COMMAND LINE AGAINST THE TABLE
                                ;       OF VALID MNEMONICS
                                ;
000046  012700  000006          1$:     MOV     #LD5LN,R0               ;NUMBER OF TABLE ENTRIES
000052  012702  000000'                 MOV     #LD5TBL,R2              ;POINT TO TABLE
```

```
 58 000056                                          CALL    SCAN                    ;MATCH AGAINST COMMAND LINE
 59 000062  103004                                  BCC     2$                      ;MATCH WAS MADE
 60 000064                                          CALL    ERR7                    ;'INVALID MEMORY MNEMONIC'
 61 000070  000167  000604                          JMP     LD5X
 62                                 ;
 63                                 ;       SAVE POINTER TO ROUTINE ASSOCIATED WITH THE MEMORY MNEMONIC
 64                                 ;       SCAN THE REMAINDER OF THE COMMAND LINE FOR CONTROL INFORMATION:
 65                                 ;
 66                                 ;       1. START ADDRESS ONLY, REPEAT PROMPT FOR EACH MEMORY LOCATION
 67                                 ;          FROM THE START ADDRESS FORWARD. NB. END OF MEMORY (AS DETECTED
 68                                 ;          BY THE QMAIN SUBROUTINE PDATA) OR A <CR> RESPONSE TO THE PROMPT
 69                                 ;          WILL TERMINATE THE LOAD.
 70                                 ;          SP>LD QX 0
 71                                 ;       2. START ADDRESS, LOOP INDICATOR. LOOP ON THE LOADING OF
 72                                 ;          THIS ONE MEMORY LOCATION ONLY.
 73                                 ;          SP>LD QX 0 L
 74                                 ;       3. START ADDRESS, END ADDRESS, NO LOOP, ISSUE ONLY ONE
 75                                 ;          PROMPT AND FILL MEMORY (BETWEEN START AND END ADDRESSES)
 76                                 ;          WITH THIS VALUE.
 77                                 ;          SP>LD QW 0 7
 78                                 ;       4. START ADDRESS, END ADDRESS, LOOP INDICATOR, ISSUE ONLY ONE
 79                                 ;          PROMPT AND LOOP ON THE LOADING OF MEMORY (BETWEEN START
 80                                 ;          AND END ADDRESSES) WITH THIS VALUE.
 81                                 ;          SP>LD QW 0 7 L
 82                                 ;
 83                                 ;
 84 000074  010167  000000G·        2$:     MOV     R1,RTNPT                ;SAVE POINTER
 85 000100                                  CALL    FIND                    ;LOCATE START ADDRESS IN COMMAND LINE
 86 000104  103004                          BCC     3$                      ;OK, CONTINUE
 87 000106                                  CALL    ERR4                    ;'MISSING OPERAND'
 88 000112  000167  000562                  JMP     LD5X                    ;EXIT
 89 000116                          3$:     CALL    PACK                    ;CONVERT COMMAND LINE VALUE TO BINARY
 90 000122  103004                          BCC     4$                      ;CONVERSION SUCCESSFUL
 91 000124                                  CALL    ERR5                    ;'INVALID NUMERIC VALUE'
 92 000130  000167  000544                  JMP     LD5X
 93                                 ;
 94 000134  016767  000000G· 000000G· 4$:   MOV     BINWD,MSTRT             ;SAVE LOADING START ADDRESS
 95 000142  016767  000000G· 000000G·       MOV     BINWD,MSTR2             ;SAVE IT TWICE (FOR REFRESH ON LOOP)
 96 000150  012767  177777  000000G·       MOV     #-1,MEND                ;INIT END ADDRESS
 97                                 ;
 98                                 ;       START ADDRESS HAS BEEN FOUND, SCAN FOR END ADDRESS OR LOOP
 99                                 ;       INDICATOR (CONDITIONS 2, 3 ABOVE). IF THERE IS NOTHING
100                                 ;       FURTHER IN THE COMMAND LINE, CONDITION 1 IS IN EFFECT.
101                                 ;
102 000156                                 CALL    FIND                    ;SCAN COMMAND LINE
103 000162  103004                         BCC     5$                      ;SOMETHING THERE
104 000164  052767  000000G· 000000G·      BIS     #RP,BASE                ;SIGNAL TO REPEAT PROMPT
105 000172  000437                         BR      9$                      ;JUMP TO RTN
106                                ;
107 000174  122711  000114          5$:    CMPB    #'L,(R1)                ;LOOP INDICATOR
108 000200  001006                         BNE     6$                      ;NO, MUST BE USPER ADDRESS
109 000202  016767  000000G· 000000G·      MOV     MSTRT,MEND              ;SET END ADDR = START ADDR
110 000210                                 CALL    HANG                    ;HOW TO STOP LOOP
111 000214  000426                         BR      9$                      ;JUMP TO RTN
112                                ;
113 000216                          6$:    CALL    PACK                    ;CONVERT UPPER ADDRESS
114 000222  103004                         BCC     7$                      ;OK, CONTINUE
```

```
115 000224                                          CALL    ERR5                    ;INVALID NUMERIC
116 000230  000167  000444                          JMP     LD5X                    ;EXIT
117                                 ;
118                                 ;       SAVE END ADDRESS (BINARY)
119                                 ;       CHECK FOR LOOP INDICATOR AFTER END ADDRESS (CONDITION 4)
120                                 ;
121 000234  016767  000000G 000000G 7$:     MOV     BINWD,MEND              ;SET UP END ADDRESS
122 000242                                  CALL    FIND                    ;CHECK FOR LOOP INDICATOR
123 000246  103411                          BCS     9$                      ;NO LOOP
124 000250  122711  000114                  CMPB    #'L,(R1)                ;CORRECT INDICATOR
125 000254  001404                          BEQ     8$                      ;YES, CONTINUE
126 000256                                  CALL    ERR11           ;LOOP OPTION ERROR
127 000262  000167  000412                  JMP     LD5X
128 000266                          8$:     CALL    HANG                    ;HOW TO STOP LOOP
129                                 ;
130 000272  016701  000000G         9$:     MOV     RTNPT,R1                ;POINT TO ROUTINE
131 000276  000171  000000                  JMP     @(R1)                   ;JUMP TO ROUTINE
132                                 ;
133                                 ;
134                                 ;       LOAD QEX MEMORY
135                                 ;
136 000302  012767  000004  000000G LD5QX:: MOV     #S$QX,CODE              ;SET MEMORY SELECT CODE = QEX
137 000310  000420                          BR      LD5Q
138                                 ;
139                                 ;       LOAD SDLB PAGE MEMORY
140                                 ;
141 000312  012767  000005  000000G LD5QB:: MOV     #S$QB,CODE              ;SET MEMORY SELECT CODE = SDLB PAGE
142 000320  000414                          BR      LD5Q
143                                 ;
144                                 ;       LOAD QLB REFERENCE PAGE
145                                 ;
146 000322                          LD5QR::
147 000322  012767  000006  000000G         MOV     #S$QR,CODE              ;SET MEMORY SELECT CODE = REFERENCE PAGE
148 000330  000410                          BR      LD5Q
149                                 ;
150                                 ;       LOAD SUBREAD MEMORY
151                                 ;
152 000332                          LD5SR::
153 000332  012767  000007  000000G         MOV     #S$SR,CODE
154 000340  016746  000000G                 MOV     SRHIGH,-(SP)            ;SUPPLY UPPER MEMORY LIMIT
155 000344  016746  000000G                 MOV     SRLOW,-(SP)             ;LOWER LIMIT
156 000350  000404                          BR      LD5GO
157 000352                          LD5Q:
158 000352  016746  000000G                 MOV     SQHIGH,-(SP)            ;SUPPLY UPPER MEMORY LIMIT
159 000356  016746  000000G                 MOV     SQLOW,-(SP)             ;LOWER LIMIT
160 000362                          LD5GO:
161 000362
162 000366  103002                          CALL    BUFSET                  ;PREPARE FOR LOAD
163 000370  000167  000304                  BCC     1$                      ;OK, CONTINUE
164                                         JMP     LD5X                    ;ERROR, EXIT
165 000374  032767  000000G 000000G 1$:     BIT     #RP,BASE                ;REPEAT PROMPT
166 000402  001414                          BEQ     5$                      ;NO, ONCE ONLY
167 000404                          2$:     CALL    PDATA                   ;READ DATA FROM COMMAND LINE
168 000410  103401                          BCS     3$                      ;END OF MEMORY
169 000412  102002                          BVC     4$                      ;NO <CR> RESPONSE, CONTINUE
170 000414                          3$:
171 000414  000167  000260                  JMP     LD5X
```

```
172·                                    ;
173 000420                              4$:
174 000420
175 000424  066767  000000G·000000G·            CALL·   LOD5M1                  ;LOAD·16 BIT·MEMORY·
176 000432· 000764                        ADD·    INCVAL,MSTR2·               ;BUMP·ADDRESS·
177                                       BR      2$                          ;REPEAT·
178                                     ;
179                                     ;       PROMPT·ONCE·THEN·FILL·MEMORY·
180 000434                              5$:     CALL·   PDATA·                  ;PROMPT·
181 000440  103401                        BCS·    6$                          ;END·OF·MEMORY·
182·000442· 102002·                       BVC·    7$                          ;NO·<CR> RESPONSE, CONTINUE·
183 000444                              6$:
184 000444  000167  000230                JMP·    LD5X
185                                     ;
186 000450                              7$:
187 000450                                CALL·   LOD5M1                  ;LOAD·16 BIT·MEMORY·
188                                     ;
189 000454  066767  000000G·000000G·      ADD·    INCVAL,MSTR2·           ;ADVANCE·ADDRESS·
190 000462· 026767  000000G·000000G·      CMP·    MSTR2,MEND·             ;HAS·UPPER MEMORY·LIMIT·BEEN·REACHED·
191 000470  101767                        BLOS·   7$                      ;NO, CONTINUE·
192·000472· 032767  000000G·000000G·      BIT·    #LOOP,BASE·             ;LOOP·ON·
193 000500  001002·                       BNE·    8$                      ;YES, CONTINUE·
194 000502· 000167  000172                JMP·    LD5X
195 000506  016767  000000G·000000G·8$:   MOV·    MSTRT,MSTR2·            ;REINITIALIZE·ADDRESS·
196 000514  000755                        BR      7$
```

```
                                        ;       LOAD SIDMEM (MULTIPLE PAGES)
                                        ;
000516                                  LD5S1::
000516  012767  000010  000000G                 MOV     #S$S1,CODE              ;SELECT SIDMEM 1
000524  000403                                  BR      LD5S
000526                                  LD5S2::
000526  012767  000014  000000G                 MOV     #S$S2,CODE              ;SELECT SIDMEM 2
000534                                  LD5S:
000534  016746  000000G                         MOV     SDHIGH,-(SP)            ;SUPPLY UPPER MEMORY LIMIT
000540  016746  000000G                         MOV     SDLOW,-(SP)             ;LOWER LIMIT
000544                                          CALL    BUFS3                   ;PREPARE FOR LOAD
000550  103002                                  BCC     1$                      ;OK, CONTINUE
000552  000167  000122                          JMP     LD5X                    ;ERROR, EXIT
                                        ;
000556  032767  000000G 000000G         1$:     BIT     #RP,BASE                ;REPEAT PROMPT
000564  001414                                  BEQ     5$                      ;NO, ONCE ONLY
000566                                  2$:     CALL    PDATA                   ;READ DATA FROM COMMAND LINE
000572  103401                                  BCS     3$                      ;END OF MEMORY
000574  102002                                  BVC     4$                      ;NO <CR> RESPONSE, CONTINUE
000576                                  3$:
000576  000167  000076                          JMP     LD5X
                                        ;
000602                                  4$:
000602                                          CALL    LOD5M3                  ;LOAD 48 BIT MEMORY
                                        ;
000606  066767  000000G 000000G                 ADD     INCVAL,MSTR2            ;BUMP ADDRESS
000614  000764                                  BR      2$                      ;REPEAT
                                        ;
                                        ;       PROMPT ONCE THEN FILL MEMORY
                                        ;
000616                                  5$:     CALL    PDATA                   ;PROMPT
000622  103401                                  BCS     6$                      ;END OF MEMORY
000624  102002                                  BVC     7$                      ;NO <CR> RESPONSE, CONTINUE
000626                                  6$:
000626  000167  000046                          JMP     LD5X
                                        ;
000632                                  7$:
000632                                          CALL    LOD5M3                  ;LOAD 48 BIT MEMORY
                                        ;
000636  066767  000000G 000000G                 ADD     INCVAL,MSTR2            ;ADVANCE ADDRESS
000644  026767  000000G 000000G                 CMP     MSTR2,MEND              ;HAS UPPER MEMORY LIMIT BEEN REACHED
000652  101767                                  BLOS    7$                      ;NO, CONTINUE
000654  032767  000000G 000000G                 BIT     #LOOP,BASE              ;LOOP ON
000662  001002                                  BNE     8$                      ;YES, CONTINUE
000664  000167  000010                          JMP     LD5X
000670  016767  000000G 000000G         8$:     MOV     MSTRT,MSTR2             ;REINITIALIZE ADDRESS
000676  000755                                  BR      7$
                                        ;
                                        ;
000700                                  LD5X:
000700  012746  000000                          MOV     #S$CLR,-(SP)            ;CLEAR SP
000704                                          CALL    SPCR                    ;WRITE TO CONTROL REG
000710  042767  000000G 000000G                 BIC     #RP,BASE                ;CLEAR PROMPT FLAG
000716                                          CALL    KILL                    ;KILL AST (IF THERE WAS ONE)
000722  000167  000000G                         JMP     SPSXX
```

```
                                        ;
                                        ;       LOAD·SINGLE·WORD·MEMORY·
                                        ;
000726                                  LOD5M1:
000726  012746  000001                          MOV·    #S$LA,-(SP)             ;ADDRESS·SELECT·
000732·                                         CALL·   SPCR                    ;SEND·TO·SP·CONTROL·REG·
000736  016746  000000G·                        MOV·    MSTR2,-(SP)             ;ACTUAL·ADDRESS·
000742·                                         CALL·   LBSP                    ;SEND·TO·SP·
000746  016746  000000G·                        MOV·    CODE,-(SP)              ;SELECT·MEMORY·
000752·                                         CALL·   SPCR
000756  016746  000000G·                        MOV·    DATA1,-(SP)             ;DATA·WORD FOR·MEMORY·
000762·                                         CALL·   LBSP                    ;SEND·DATA TO·SP·
000766                                          RETURN·
                                        ;
                                        ;       LOAD·3 WORD·MEMORY·
                                        ;
000770                                  LOD5M3:
000770                                          SAVE·   CODE,DATA1
001000                                          CALL·   LOD5M1                  ;LOAD·WORD A·
001004  005267  000000G·                        INC·    CODE                    ;SELECT·WORD·B·
001010  016767  000000G· 000000G·               MOV·    DATA2,DATA1             ;SET·UP·DATA1
001016                                          CALL·   LOD5M1                  ;LOAD·WORD B·
001022· 005267  000000G·                        INC·    CODE                    ;SELECT·WORD·C·
001026  016767  000000G· 000000G·               MOV·    DATA3,DATA1             ;SET·UP·DATA1
001034                                          CALL·   LOD5M1                  ;LOAD·WORD C·
001040                                          RESTOR· CODE,DATA1
001050                                          RETURN·
                                        ;
        000001                                  .END·
```

```
ALUCKE= 040000
ALUOE = 004000
A01   = 010000
BASE  = ****** GX
BINWD = ****** GX
BITVAL= 000000
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BUFSET= ****** GX
BUFS3 = ****** GX
BYTE0 = 000000
BYTE1 = 000001
BYTE10= 000012
BYTE11= 000013
BYTE12= 000014
BYTE13= 000015
BYTE14= 000016
BYTE15= 000017
BYTE16= 000020
BYTE17= 000021
BYTE18= 000022
BYTE19= 000023
BYTE2 = 000002
BYTE20= 000024
BYTE21= 000025
BYTE22= 000026
BYTE23= 000027
BYTE24= 000030
BYTE25= 000031
BYTE26= 000032
BYTE27= 000033
BYTE28= 000034
BYTE29= 000035
BYTE3 = 000003
BYTE30= 000036
BYTE31= 000037
BYTE32= 000040
BYTE33= 000041
BYTE34= 000042
BYTE35= 000043
BYTE36= 000044
BYTE37= 000045
BYTE38= 000046
BYTE39= 000047
BYTE4 = 000004
BYTE40= 000050
BYTE41= 000051
BYTE42= 000052
BYTE43= 000053
BYTE44= 000054
BYTE45= 000055
BYTE46= 000056
BYTE47= 000057
BYTE48= 000060
BYTE49= 000061
BYTE5 = 000005
BYTE50= 000062
BYTE51= 000063
BYTE52= 000064
BYTE53= 000065
BYTE54= 000066
BYTE55= 000067
BYTE56= 000070
BYTE57= 000071
BYTE58= 000072
BYTE59= 000073
BYTE6 = 000006
BYTE60= 000074
BYTE61= 000075
BYTE62= 000076
BYTE63= 000077
BYTE64= 000100
BYTE65= 000101
BYTE66= 000102
BYTE67= 000103
BYTE68= 000104
BYTE69= 000105
BYTE7 = 000007
BYTE70= 000106
BYTE71= 000107
BYTE72= 000110
BYTE73= 000111
BYTE74= 000112
BYTE75= 000113
BYTE76= 000114
BYTE77= 000115
BYTE78= 000116
BYTE79= 000117
BYTE8 = 000010
BYTE80= 000120
BYTE81= 000121
BYTE82= 000122
BYTE83= 000123
BYTE84= 000124
BYTE85= 000125
BYTE86= 000126
BYTE87= 000127
BYTE88= 000130
BYTE89= 000131
BYTE9 = 000011
BYTE90= 000132
BYTE91= 000133
BYTE92= 000134
BYTE93= 000135
BYTE94= 000136
BYTE95= 000137
BYTE96= 000140
BYTE97= 000141
BYTE98= 000142
BYTE99= 000143
BYTVAL= 000144
CBKALL= 001000
CBKCLK= 000400
CNOBRE= 100000
CODE  = ****** GX
CPCCEN= 010000
CPREAD= 040000
CPWRTE= 020000
CSADRD= 000004
CSEQCI= 100000
CSOE  = 000040
CSWRTE= 000100
DATA1 = ****** GX
DATA2 = ****** GX
DATA3 = ****** GX
DBR.RD= 000001
DB$CPP= 001457
DB$SPT= 000026
DB$TPC= 000023
DISPGS= 100000
DMAAWR= 000005
DMARRD= 000003
DMARWR= 000004
ENBR  = 010000
ERR11 = ****** GX
ERR4  = ****** GX
ERR5  = ****** GX
ERR7  = ****** GX
FIND  = ****** GX
HANG  = ****** GX
INCVAL= ****** GX
KILL  = ****** GX
LBSP  = ****** GX
LD5     000030RG
LD5GO   000362R
LD5LN = 000006 G
LD5Q    000352R
LD5QB   000312RG
LD5QR   000322RG
LD5QX   000302RG
LD5S    000534R
LD5SR   000332RG
LD5S1   000516RG
LD5S2   000526RG
LD5TBL  000000RG
LD5X    000700R
LOC.EN= 000100
LOC.WA= 040000
LOC.WB= 100000
LOD5M1  000726R     002
LOD5M3  000770R     002
LOOP  = ****** GX
MAREN1= 000001
MAREN2= 004000
MARLOD= 010000
MAROUT= 000002
MAR.LO= 002000
MAR.UO= 000040
MBKALL= 001000
MBKCLK= 000400
MEND  = ****** GX
MMADRD= 000100
MMLEFT= 000002
MMOE  = 000004
MMWRTE= 000010
MNOBRE= 100000
MREN1 = 000001
MREN2 = 020000
MSTRT = ****** GX
MSTR2 = ****** GX
MSYN  = 000040
N     = 000144
PACK  = ****** GX
PDATA = ****** GX
PLB   = 000010
PLC   = 000020
PLD   = 000030
PLRWR = 000200
PLR.EN= 000200
QR$CR1= 176420
QR$CR2= 176422
QR$LBR= 176424
Q$ATTN= 000100
Q$BCL = 000001
Q$CCCP= 000040
Q$CHB = 000400
Q$CHRL= 000200
Q$CLR = 000040
Q$CNC = 030000
Q$CP  = 000060
002 Q$CPCC= 000010
002 Q$CP2 = 000260
Q$CSC = 010000
002 Q$CSEL= 000360
002 Q$CSET= 000002
002 Q$CSP = 020000
002 Q$DMA = 000001
002 Q$ENBK= 040000
002 Q$ENOP= 020000
002 Q$FAL = 004000
002 Q$FC  = 000045
002 Q$FO  = 000044
002 Q$FP  = 000046
Q$HBE = 000002
Q$ICP = 000006
Q$IHB = 000003
002 Q$IHRL= 000002
002 Q$IMRP= 000007
Q$LBD = 001000
Q$LBDP= 001001
Q$LBP = 000001
Q$LDCD= 000003
Q$LDMD= 000004
Q$LDPP= 002000
Q$LHP = 010000
Q$MNC = 140000
Q$MR  = 000052
Q$MRP = 000040
Q$MRP2= 000240
Q$MSC = 040000
Q$MSET= 000004
Q$MSP = 100000
Q$NCLK= 176000
Q$PP  = 000100
Q$PPSW= 000320
Q$PP2 = 000300
Q$QHLT= 000013
Q$QL  = 000043
Q$QLA = 000053
Q$QLB = 000054
Q$QLR = 000001
Q$QW  = 000042
Q$RDCD= 000005
Q$RDMD= 000006
Q$REBK= 001000
Q$RNC = 006000
Q$RSC = 004000
Q$RSET= 000010
Q$SM  = 100000
Q$SP  = 000120
Q$SP2 = 000340
RGQ.EN= 000200
RGQ.VA= 020000
RP    = ****** GX
RTNPT = ****** GX
SCAN  = ****** GX
SDHIGH= ****** GX
SDLOW = ****** GX
SEQ.CF= 000010
SPCR  = ****** GX
SPSXX = ****** GX
SQHIGH= ****** GX
SQLOW = ****** GX
SRHIGH= ****** GX
SRLOW = ****** GX
S$CLR = 000000
S$LA  = 000001
S$QB  = 000005
S$QR  = 000006
S$QX  = 000004
S$SR  = 000007
```

```
S$S1 = 000010
S$S2 = 000014
TD$CTR = 176370
TD$CTW = 176360
TD$INL = 004000
TD$MEM = 000270
TD$OAR = 176344
TD$OTR = 176346
TD$QRD = 000274
TD$SW = 176376
TD$TAR = 176372
TD$TAW = 176362
TD$TDR = 176374
TD$TDW = 176364
T$AD = 000020
T$BA = 000002
T$BD = 000010
T$BSO = 100000
T$BT = 000020
T$BTAR = 000030
T$BTD = 002000
T$CD = 000100
T$CLK = 002000
T$DISK = 000200
T$DRD = 000004
T$EMEM = 010000
T$FSAA = 000000
T$FSAB = 000004
T$FSAC = 000014
T$FSB2 = 000010
T$IB = 000026
T$IBAR = 000024
T$IBE = 020000
T$IBF = 040000
T$ICD = 000040
T$MODE = 004000
T$OB = 000036
T$OBE = 004000
T$OBF = 010000
T$OBRA = 000034
T$OBWA = 000032
T$OUTA = 100000
T$RBDO = 000200
T$RNB = 000040
T$RSET = 040000
T$SC = 000022
T$SCLK = 020000
T$SEG1 = 000000
T$SEG2 = 000001
T$SEG3 = 000002
T$SO = 000001
T$UBUS = 100000
T$1CLK = 000400
T$8BEN = 000020
UBD.IN = 000020
WORD0 = 000000
WORD1 = 000002
WORD10 = 000024
WORD11 = 000026
WORD12 = 000030
WORD13 = 000032
WORD14 = 000034
WORD15 = 000036
WORD16 = 000040
WORD17 = 000042
WORD18 = 000044
WORD19 = 000046
WORD2 = 000004
WORD20 = 000050
WORD21 = 000052
WORD22 = 000054
WORD23 = 000056
WORD24 = 000060
WORD25 = 000062
WORD26 = 000064
WORD27 = 000066
WORD28 = 000070
WORD29 = 000072
WORD3 = 000006
WORD30 = 000074
WORD31 = 000076
WORD32 = 000100
WORD33 = 000102
WORD34 = 000104
WORD35 = 000106
WORD36 = 000110
WORD37 = 000112
WORD38 = 000114
WORD39 = 000116
WORD4 = 000010
WORD40 = 000120
WORD41 = 000122
WORD42 = 000124
WORD43 = 000126
WORD44 = 000130
WORD45 = 000132
WORD46 = 000134
WORD47 = 000136
WORD48 = 000140
WORD49 = 000142
WORD5 = 000012
WORD50 = 000144
WORD51 = 000146
WORD52 = 000150
WORD53 = 000152
WORD54 = 000154
WORD55 = 000156
WORD56 = 000160
WORD57 = 000162
WORD58 = 000164
WORD59 = 000166
WORD6 = 000014
WORD60 = 000170
WORD61 = 000172
WORD62 = 000174
WORD63 = 000176
WORD64 = 000200
WORD65 = 000202
WORD66 = 000204
WORD67 = 000206
WORD68 = 000210
WORD69 = 000212
WORD7 = 000016
WORD70 = 000214
WORD71 = 000216
WORD72 = 000220
WORD73 = 000222
WORD74 = 000224
WORD75 = 000226
WORD76 = 000230
WORD77 = 000232
WORD78 = 000234
WORD79 = 000236
WORD8 = 000020
WORD80 = 000240
WORD81 = 000242
WORD82 = 000244
WORD83 = 000246
WORD84 = 000250
WORD85 = 000252
WORD86 = 000254
WORD87 = 000256
WORD88 = 000260
WORD89 = 000262
WORD9 = 000022
WORD90 = 000264
WORD91 = 000266
WORD92 = 000270
WORD93 = 000272
WORD94 = 000274
WORD95 = 000276
WORD96 = 000300
WORD97 = 000302
WORD98 = 000304
WORD99 = 000306
WRDVAL = 000310
XTREAD = 001000
XTWRTE = 000400

. ABS.  000000  000
        000000  001
SPLD    001052  002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  3456 WORDS  ( 14 PAGES)
DYNAMIC MEMORY:  4916 WORDS  ( 18 PAGES)
ELAPSED TIME:  00:00:46
SPLD,SPLD/-SP=[20,1]IM,[20,1]SPLD
```

```
                                        .TITLE  SPPR
000000                                  .PSECT  SPPR
                                ;
                                ;
                                ;       HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
                                ;       SUBDOCUMENT PROCESSOR TEST ROUTINES
                                ;
                                ;       COMMANDS:
                                ;       PR      PRINT SP MEMORIES
                                ;
                                ;       ONCE A COMMAND HAS BBEN EXECUTED (OR AN ERROR ENCOUNTERED)
                                ;       THIS MODULE RETURNS CONTROL TO THE MODULE SP AT LOCATION
                                ;       'SPXX'.
                                ;
                                ;
                                ;
                                        .MCALL  WTSE$S,CLEF$S
                                ;
                                ;
                                ;       TABLE OF VALID SP MEMORY MNEMONICS AND ADDRESSES OF THEIR
                                ;       ASSOCIATED ROUTINES
                                ;
000000                          PR5TBL::
000000  121     130                     .ASCII  /QX/                    ;QEX MEMORY
000002  000316'                         .WORD   PR5QX
000004  123     122                     .ASCII  /SR/                    ;SUBREAD MEMORY
000006  000346'                         .WORD   PR5SR
000010  123     106                     .ASCII  /SF/                    ;SDLB REFERENCE PAGE
000012  000336'                         .WORD   PR5QR
000014  123     060                     .ASCII  /S0/                    ;SDLB PAGE
000016  000326'                         .WORD   PR5QB
000020  123     061                     .ASCII  /S1/                    ;SIDMEM1
000022  000470'                         .WORD   PR5S1
000024  123     062                     .ASCII  /S2/                    ;SIDMEM2
000026  000500'                         .WORD   PR5S2
000030                          PR5LN   ==      <.-PR5TBL>
                                ;
                                ;
                                ;
                                ;       PRINT
                                ;       PERFORM THIRD LEVEL PARSING
                                ;       EG. IN THE COMMAND:
                                ;       SP>PR QX 0
                                ;       PARSE THE 'QX'
                                ;
                                ;
000030                          SPPR5::
000030
000034  103004                        CALL    FIND                    ;LOCATE MEMORY MNEMONIC IN COMMAND LINE
000036                                  BCC     1$                      ;OK, CONTINUE
000042  000167  000614                 CALL    ERR4                    ;'MISSING OPERAND'
                                        JMP     PR5X                    ;EXIT
                                ;
                                ;       MATCH THE MNEMONIC IN THE COMMAND LINE AGAINST THE TABLE
                                ;       OF VALID MNEMONICS
                                ;
000046  012700  000030          1$:     MOV     #PR5LN,R0               ;NUMBER OF TABLE ENTRIES
000052  012702  000000'                 MOV     #PR5TBL,R2              ;POINT TO TABLE
```

```
 58 000056
 59 000062  103004                          CALL    SCAN                    ;MATCH AGAINST COMMAND LINE.
 60 000064                                  BCC     2$                      ;MATCH WAS MADE.
 61 000070  000167  000566                  CALL    ERR7                    ;'INVALID MEMORY MNEMONIC'
 62                                         JMP     PR5X
 63                                 ;
 64                                 ;       SAVE POINTER TO ROUTINE ASSOCIATED WITH THE MEMORY MNEMONIC.
 65                                 ;       SCAN THE REMAINDER OF THE COMMAND LINE FOR CONTROL INFORMATION:
 66                                 ;
 67                                 ;       1. START ADDRESS ONLY. REPEAT PROMPT FOR EACH MEMORY LOCATION
 68                                 ;          FROM THE START ADDRESS FORWARD. NB. END OF MEMORY (AS DETECTED
 69                                 ;          BY THE QMAIN SUBROUTINE PDATA) OR A <CR> RESPONSE TO THE PROMPT
 70                                 ;          WILL TERMINATE THE LOAD.
 71                                 ;          SP>PR QX 0
 72                                 ;       2. START ADDRESS, LOOP INDICATOR. LOOP ON THE LOADING OF
 73                                 ;          THIS ONE MEMORY LOCATION ONLY.
 74                                 ;          SP>PR QX 0 L
 75                                 ;       3. START ADDRESS, END ADDRESS, NO LOOP. ISSUE ONLY ONE
 76                                 ;          PROMPT AND FILL MEMORY (BETWEEN START AND END ADDRESSES)
 77                                 ;          WITH THIS VALUE.
 78                                 ;          SP>PR QW 0 7
 79                                 ;       4. START ADDRESS, END ADDRESS, LOOP INDICATOR. ISSUE ONLY ONE
 80                                 ;          PROMPT AND LOOP ON THE LOADING OF MEMORY (BETWEEN START
 81                                 ;          AND END ADDRESSES) WITH THIS VALUE.
 82                                 ;          SP>PR QW 0 7 L
 83                                 ;
 84 000074  010167  000000G         2$:     MOV     R1,RTNPT                ;SAVE POINTER.
 85 000100                                  CALL    FIND                    ;LOCATE START ADDRESS IN COMMAND LINE.
 86 000104  103004                          BCC     3$                      ;OK, CONTINUE.
 87 000106                                  CALL    ERR4                    ;'MISSING OPERAND'
 88 000112  000167  000544                  JMP     PR5X                    ;EXIT.
 89 000116                          3$:     CALL    PACK                    ;CONVERT COMMAND LINE VALUE TO BINARY.
 90 000122  103004                          BCC     4$                      ;CONVERSION SUCCESSFUL.
 91 000124                                  CALL    ERR5                    ;'INVALID NUMERIC VALUE'
 92 000130  000167  000526                  JMP     PR5X
 93                                 ;
 94 000134  016767  000000G 000000G 4$:     MOV     BINWD,MSTRT             ;SAVE LOADING START ADDRESS.
 95 000142  016767  000000G 000000G         MOV     BINWD,MSTR2             ;SAVE IT TWICE (FOR REFRESH ON LOOP)
 96 000150  012767  177777  000000G         MOV     #-1,MEND                ;INIT END ADDRESS.
 97                                 ;
 98                                 ;       START ADDRESS HAS BEEN FOUND. SCAN FOR END ADDRESS OR LOOP
 99                                 ;       INDICATOR (CONDITIONS 2, 3 ABOVE)--IF THERE IS NOTHING
100                                 ;       FURTHER IN THE COMMAND LINE, CONDITION 1 IS IN EFFECT.
101                                 ;
102 000156                                 CALL    FIND                    ;SCAN COMMAND LINE.
103 000162  103004                         BCC     5$                      ;SOMETHING THERE.
104 000164  016767  000000G 000000G       MOV     MSTRT,MEND              ;SET END ADDR = START ADDR.
105 000172  000445                         BR      9$                      ;JUMP TO RTN.
106                                 ;
107 000174  122711  000114         5$:     CMPB    #'L,(R1)                ;LOOP INDICATOR.
108 000200  001011                         BNE     6$                      ;NO, MUST BE USPER ADDRESS.
109 000202  052767  000000G 000000G       BIS     #OUT.BASE               ;SET FLAG FOR OUTPUT CONTROL.
110 000210  016767  000000G 000000G       MOV     MSTRT,MEND              ;SET END ADDR = START ADDR.
111 000216                                 CALL    HANG                    ;HOW TO STOP LOOP.
112 000222  000431                         BR      9$                      ;JUMP TO RTN.
113                                 ;
114 000224                          6$:    CALL    PACK                    ;CONVERT BINARY ADDRESS
```

```
000230  103004                          BCC···  7$                      ;OK, CONTINUE·
000232·                                 CALL·   ERR5                    ;INVALID·NUMERIC·
000236  000167  000420                  JMP·    PR5X                    ;EXIT·
                                ;
                                ;       SAVE·END·ADDRESS·(BINARY)
                                ;       CHECK·FOR·LOOP·INDICATOR·AFTER·END·ADDRESS (CONDITION·4)
                                ;
000242· 016767  000000G·000000G· 7$:    MOV·    BINWD,MEND·             ;SET·UP·END·ADDRESS·
000250                                  CALL·   FIND                    ;CHECK·FOR·LOOP·INDICATOR·
000254  103414                          BCS·    9$                      ;NO·LOOP·
000256  122711  000114                  CMPB·   #'L,(R1)                ;CORRECT·INDICATOR·
000262· 001404                          BEQ·    8$                      ;YES, CONTINUE·
000264                                  CALL·   ERR11                   ;LOOP·OPTION·ERROR·
000270  000167  000366                  JMP·    PR5X
000274                          8$:
000274  052767  000000G·000000G·        BIS·    #OUT,HANG·              ;SET·OUTPUT·CONTROL·
000302·                                 CALL·   HANG                    ;HOW·TO·STOP·LOOP·
                                ;
000306  016701  000000G·        9$:     MOV·    RTNPT,R1                ;POINT·TO·ROUTINE·
000312· 000171  000000                  JMP·    @(R1)                   ;JUMP·TO·ROUTINE·
                                ;
                                ;
                                ;       PRINT·QEX·MEMORY·
                                ;
000316  012767  000004  000000G· PR5QX:: MOV·    #S$QX,CODE·             ;SET·MEMORY·SELECT·CODE·=·QEX·
000324  000420                          BR      PR5Q
                                ;
                                ;       PRINT·SDLB·PAGE·MEMORY
                                ;
000326  012767  000005  000000G· PR5QB:: MOV·    #S$QB,CODE·             ;SET·MEMORY·SELECT·CODE·=·SDLB PAGE·
000334  000414                          BR      PR5Q
                                ;
                                ;       PRINT·QLB·REFERENCE·PAGE·
                                ;
000336                          PR5QR::
000336  012767  000006  000000G·        MOV·    #S$QR,CODE·             ;SET·MEMORY·SELECT·CODE·=·REFERENCE·PAGE·
000344  000410                          BR      PR5Q
                                ;
                                ;       PRINT·SUBREAD·MEMORY·
                                ;
000346                          PR5SR::
000346  012767  000007  000000G·        MOV·    #S$SR,CODE·
000354  016746  000000G·                MOV·    SRHIGH,-(SP)            ;SUPPLY·UPPER·MEMORY LIMIT·
000360  016746  000000G·                MOV·    SRLOW,-(SP)             ;LOWER·LIMIT·
000364  000404                          BR      PR5GO·
000366                          PR5Q:
000366  016746  000000G·                MOV·    SQHIGH,-(SP)            ;SUPPLY·UPPER·MEMORY LIMIT·
000372· 016746  000000G·                MOV·    SQLOW,-(SP)             ;LOWER·LIMIT·
000376                          PR5GO:
000376                                  CALL·   BUFSET·                 ;PREPARE·FOR·PRINT·
000402  103002·                          BCC·    1$                      ;OK, CONTINUE·
000404  000167  000252                  JMP·    PR5X                    ;ERROR, EXIT·
                                ;
000410                          1$:
000410                                  CALL·   RD5M1                   ;READ·16-BIT·MEMORY·
000414  016767  000000G·000000G·        MOV·    DATA4,DATA1             ;GET·ITS·VALUE·
000422·                                 CALL·   PRDATA·                 ;PRINT·IT·
```

```
172                                     ;
173 000426  066767  000000G 000000G          ADD     INCVAL,MSTR2            ;BUMP ADDRESS
174 000434  026767  000000G 000000G          CMP     MSTR2,MEND              ;HAS UPPER MEMORY LIMIT BEEN REACHED
175 000442  101762                            BLOS    1$                      ;NO, CONTINUE
176 000444  032767  000000G 000000G          BIT     #LOOP,BASE              ;LOOP ON
177 000452  001002                            BNE     8$                      ;YES, CONTINUE
178 000454  000167  000202                    JMP     PR5X
179 000460  016767  000000G 000000G 8$:      MOV     MSTRT,MSTR2             ;REINITIALIZE ADDRESS
180 000466  000750                            BR      1$
```

```
                                ;       PRINT·SIDMEM (MULTIPLE PAGES)
                                ;
000470                          PR5S1::
000470  012767  000010  000000G·         MOV·    #S$S1,CODE·             ;SELECT·SIDMEM·1
000476  000403                          BR      PR5S
000500                          PR5S2::
000500  012767  000014  000000G·         MOV·    #S$S2,CODE·             ;SELECT·SIDMEM·2·
000506                          PR5S:
000506  016746  000000G·                 MOV·    SDHIGH,-(SP)            ;SUPPLY·UPPER·MEMORY LIMIT·
000512· 016746  000000G·                 MOV·    SDLOW,-(SP)             ;LOWER·LIMIT·
000516                                   CALL·   BUFS3                   ;PREPARE·FOR·READ·
000522· 103002·                          BCC·    1$                      ;OK, CONTINUE·
000524  000167  000132                   JMP·    PR5X                    ;ERROR, EXIT·
                                ;
000530                          1$:
000530                                   SAVE·   CODE
000534                                   CALL·   RD5M1                   ;READ·MEMORY·WORD·A·
000540  016767  000000G·000000G·         MOV·    DATA4,DATA1             ;SAVE·FOR·PRINTING·
000546  005267  000000G·                 INC·    CODE                    ;SELECT·WORD·B·
000552·                                  CALL·   RD5M1                   ;READ·MEMORY·WORD·B·
000556· 016767  000000G·000000G·         MOV·    DATA4,DATA2·            ;SAVE·FOR·PRINTING·
000564  005267  000000G·                 INC·    CODE                    ;SELECT·WORD·C·
000570                                   CALL·   RD5M1                   ;READ·MEMORY·WORD·C·
000574  016767  000000G·000000G·         MOV·    DATA4,DATA3             ;SAVE·FOR·PRINTING·
000602·                                  RESTOR· CODE
                                ;
000606                                   CALL·   PRDATA·                 ;PRINT·MEMORY·CONTENTS·
000612· 103002·                          BCC·    20$                     ;NOT·END·OF·MEMORY·
000614  000167  000042                   JMP·    PR5X                    ;END·OF·MEMORY·
                                ;
000620                          20$:
000620  066767  000000G·000000G·         ADD·    INCVAL,MSTR2·           ;BUMP·ADDRESS·
000626  026767  000000G·000000G·         CMP·    MSTR2,MEND·             ;HAS·UPPER MEMORY·LIMIT·BEEN·REACHED·
000634  101735                          BLOS·   1$                      ;NO, CONTINUE·
000636  032767  000000G·000000G·         BIT·    #LOOP,BASE·             ;LOOP·ON·
000644  001002·                          BNE·    8$                      ;YES, CONTINUE·
000646  000167  000010                   JMP·    PR5X
000652· 016767  000000G·000000G·8$:      MOV·    MSTRT,MSTR2·            ;REINITIALIZE·ADDRESS·
000660  000723                          BR      1$
                                ;
                                ;
000662·                         PR5X:
000662· 012746  000000                   MOV·    #S$CLR,-(SP)            ;CLEAR·SP·
000666                                   CALL·   SPCR                    ;WRITE·TO·CONTROL·REG·
000672  042767  000000C·000000G·         BIC·    #<ONCE+OUT>,BASE·       ;CLEAR·PROMPT·FLAG·
000700                                   CALL·   KILL                    ;KILL·AST·(IF·THERE·WAS·ONE)
000704  000167  000000G·                 JMP·    SPSXX·
```

```
230                                     ;
231                                     ;       PRINT·SINGLE WORD·MEMORY·
232·                                    ;
233 000710                              RD5M1:
234 000710  012746  000001                      MOV·    #S$LA,-(SP)             ;ADDRESS·SELECT·
235 000714                                      CALL·   SPCR                    ;SEND·TO·SP·CONTROL·REG·
236 000720  016746  000000G·                    MOV·    MSTR2,-(SP)             ;ACTUAL·ADDRESS·
237 000724                                      CALL·   LBSP                    ;SEND·TO·SP·
238 000730  016746  000000G·                    MOV·    CODE,-(SP)              ;SELECT·MEMORY·
239 000734                                      CALL·   SPCR
240 000740                                      CALL·   SPLB                    ;DO·SP·TO·LOD·BUS·
241 000744  012667  000000G·                    MOV·    (SP)+,DATA4             ;FETCH·DATA·
242·000750                                      RETURN·
243                                     ;
244         000001                              .END·
```

SYMBOL TABLE

```
ALUCKE= 040000
ALUOE = 010000
A01   = 010000
BASE  = ****** GX
BINWD = ****** GX
BITVAL= 000000
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BUFSET= ****** GX
BUFS3 = ****** GX
BYTE0 = 000000
BYTE1 = 000001
BYTE10= 000012
BYTE11= 000013
BYTE12= 000014
BYTE13= 000015
BYTE14= 000016
BYTE15= 000017
BYTE16= 000020
BYTE17= 000021
BYTE18= 000022
BYTE19= 000023
BYTE2 = 000002
BYTE20= 000024
BYTE21= 000025
BYTE22= 000026
BYTE23= 000027
BYTE24= 000030
BYTE25= 000031
BYTE26= 000032
BYTE27= 000033
BYTE28= 000034
BYTE29= 000035
BYTE3 = 000003
BYTE30= 000036
BYTE31= 000037
BYTE32= 000040
BYTE33= 000041
BYTE34= 000042
BYTE35= 000043
BYTE36= 000044
BYTE37= 000045
BYTE38= 000046
BYTE39= 000047
BYTE4 = 000004
BYTE40= 000050
BYTE41= 000051
BYTE42= 000052
BYTE43= 000053
BYTE44= 000054
BYTE45= 000055
BYTE46= 000056
BYTE47= 000057
BYTE48= 000060
BYTE49= 000061
BYTE5 = 000005
BYTE50= 000062
BYTE51= 000063
BYTE52= 000064
BYTE53= 000065
BYTE54= 000066
BYTE55= 000067
BYTE56= 000070
BYTE57= 000071
BYTE58= 000072
BYTE59= 000073
BYTE6 = 000006
BYTE60= 000074
BYTE61= 000075
BYTE62= 000076
BYTE63= 000077
BYTE64= 000100
BYTE65= 000101
BYTE66= 000102
BYTE67= 000103
BYTE68= 000104
BYTE69= 000105
BYTE7 = 000007
BYTE70= 000106
BYTE71= 000107
BYTE72= 000110
BYTE73= 000111
BYTE74= 000112
BYTE75= 000113
BYTE76= 000114
BYTE77= 000115
BYTE78= 000116
BYTE79= 000117
BYTE8 = 000010
BYTE80= 000120
BYTE81= 000121
BYTE82= 000122
BYTE83= 000123
BYTE84= 000124
BYTE85= 000125
BYTE86= 000126
BYTE87= 000127
BYTE88= 000130
BYTE89= 000131
BYTE9 = 000011
BYTE90= 000132
BYTE91= 000133
BYTE92= 000134
BYTE93= 000135
BYTE94= 000136
BYTE95= 000137
BYTE96= 000140
BYTE97= 000141
BYTE98= 000142
BYTE99= 000143
BYTVAL= 000144
CBKALL= 001000
CBKCLK= 000400
CNOBRE= 100000
CODE  = ****** GX
CPCCEN= 010000
CPREAD= 040000
CPWRTE= 020000
CSADRD= 000004
CSEQCI= 100000
CSOE  = 000040
CSWRTE= 000100
DATA1 = ****** GX
DATA2 = ****** GX
DATA3 = ****** GX
DATA4 = ****** GX
DBR.RD= 000001
DB$CPP= 001457
DB$SPT= 000026
DB$TPC= 000023
DISPGS= 100000
DMAAWR= 000005
DMARRD= 000003
DMARWR= 000004
ENBR  = 010000
ERR1  = ****** GX
ERR4  = ****** GX
ERR5  = ****** GX
ERR7  = ****** GX
FIND  = ****** GX
HANG  = ****** GX
INCVAL= ****** GX
KILL  = ****** GX
LBSP  = ****** GX
LOC.EN= 000100
LOC.WA= 040000
LOC.WB= 100000
LOOP  = ****** GX
MAREN1= 000001
MAREN2= 004000
MARLOD= 010000
MAROUT= 000002
MAR.LO= 002000
MAR.OU= 000040
MBKALL= 001000
MBKCLK= 000400
MEND  = ****** GX
MMADRD= 000100
MMLEFT= 000002
MMOE  = 000004
MMWRTE= 000010
MNOBRE= 100000
MREN1 = 000001
MREN2 = 020000
MSTRT = ****** GX
MSTR2 = ****** GX
MSYN  = 000040
N     = 000144
ONCE  = ****** GX
OUT   = ****** GX
PACK  = ****** GX
PLB   = 000010
PLC   = 000020
PLD   = 000030
PLRWR = 000200
PLR.EN= 000200
PRDATA= ****** GX
PR5GO   000376R
PR5LN = 000030 G
PR5Q    000366R
PR5QB   000326RG
PR5QR   000336RG
PR5QX   000316RG
PR5S    000506R
PR5SR   000346RG
PR5S1   000470RG
PR5S2   000500RG
PR5TBL  000000RG
PR5X    000662R
QR$CR1= 176420
QR$CR2= 176422
QR$LBR= 176424
Q$ATTN= 000100
Q$BCL = 000001
Q$CCCP= 000040
Q$CHB = 000400
Q$CHRL= 000200
Q$CLR = 000040
Q$CNC = 030000
Q$CP  = 000060
Q$CPCC= 000010
Q$CP2 = 000260
Q$CSC = 010000
Q$CSEL= 000360
Q$CSET= 000002
Q$CSP = 020000
Q$DMA = 000001
Q$ENBK= 040000
Q$ENOP= 020000
Q$FAL = 004000
Q$FC  = 000045
Q$FO  = 000044
Q$FP  = 000046
Q$HBF = 000002
Q$ICP = 000006
Q$IHB = 000003
Q$IHRL= 000002
Q$IMRP= 000007
Q$LBD = 001000
Q$LBDP= 001001
Q$LBP = 000001
Q$LDCD= 000003
Q$LDMD= 000004
Q$LDPP= 002000
Q$LHP = 010000
Q$MNC = 140000
Q$MR  = 000052
Q$MRP = 000040
Q$MRP2= 000240
Q$MSC = 040000
Q$MSET= 000004
Q$MSP = 100000
Q$NCLK= 176000
Q$PP  = 000100
Q$PPSW= 000320
Q$PP2 = 000300
Q$QHLT= 000013
Q$QL  = 000043
Q$QLA = 000053
Q$QLB = 000054
Q$QLR = 000001
Q$QW  = 000042
Q$RDCD= 000005
Q$RDMD= 000006
Q$REBK= 001000
Q$RNC = 006000
Q$RSC = 004000
Q$RSET= 000010
Q$SM  = 100000
Q$SP  = 000120
Q$SP2 = 000340
RD5M1   000710R    002
RGQ.EN= 000200
RGQ.VA= 020000
RTNPT = ****** GX
SCAN  = ****** GX
SDHIGH= ****** GX
SDLOW = ****** GX
SEQ.CI= 000010
SPCR  = ****** GX
SPLB  = ****** GX
SPPR5   000030RG   002
SPSXX = ****** GX
SQHIGH= ****** GX
SQLOW = ****** GX
SRHIGH= ****** GX
SRLOW = ****** GX
S$CLR = 000000
S$LA  = 000001
S$QB  = 000005
S$QR  = 000006
```

```
SPPR·   M[illegible]0·M1110  27-MAR-80 15:33  PAGE 7-2
SYMBOL·TA[illegible]·
```

```
S$QX·  =·000004          T$IB·  =·000026          WORD15=·000036          WORD44=·000130          WORD73=·000222
S$SR·  =·000007          T$IBAR=·000024          WORD16=·000040          WORD45=·000132·         WORD74=·000224
S$S1   =·000010          T$IBE·=·020000          WORD17=·000042·         WORD46=·000134          WORD75=·000226
S$S2·  =·000014          T$IBF·=·040000          WORD18=·000044          WORD47=·000136          WORD76=·000230
TD$CTR=·176370           T$ICD·=·000040          WORD19=·000046          WORD48=·000140          WORD77=·000232
TD$CTW=·176360           T$MODE=·004000          WORD2·=·000004          WORD49=·000142·         WORD78=·000234
TD$INL=·004000           T$OB·  =·000036         WORD20=·000050          WORD5  =·000012·        WORD79=·000236
TD$MEM=·000270           T$OBE·=·004000          WORD21=·000052·         WORD50=·000144          WORD8  =·000020
TD$OAR=·176344           T$OBF·=·010000          WORD22=·000054          WORD51=·000146          WORD80=·000240
TD$OTR=·176346           T$OBRA=·000034          WORD23=·000056          WORD52=·000150          WORD81=·000242
TD$QRD=·000274           T$OBWA=·000032·         WORD24=·000060          WORD53=·000152·         WORD82=·000244
TD$SW·=·176376           T$OUTA=·100000          WORD25=·000062·         WORD54=·000154          WORD83=·000246
TD$TAR=·176372·          T$RBDO=·000200          WORD26=·000064          WORD55=·000156          WORD84=·000250
TD$TAW=·176362·          T$RNB·=·000040          WORD27=·000066          WORD56=·000160          WORD85=·000252
TD$TDR=·176374           T$RSET=·040000          WORD28=·000070          WORD57=·000162·         WORD86=·000254
TD$TDW=·176364           T$SC·  =·000022·        WORD29=·000072·         WORD58=·000164          WORD87=·000256
T$AD·  =·000020          T$SCLK=·020000          WORD3  =·000006         WORD59=·000166          WORD88=·000260
T$BA·  =·000002·         T$SEG1=·000000          WORD30=·000074          WORD6  =·000014         WORD89=·000262
T$BD·  =·000010          T$SEG2=·000001          WORD31=·000076          WORD60=·000170          WORD9  =·000022
T$BSO·=·100000           T$SEG3=·000002·         WORD32=·000100          WORD61=·000172·         WORD90=·000264
T$BT·  =·000020          T$SO·  =·000001         WORD33=·000102·         WORD62=·000174          WORD91=·000266
T$BTAR=·000030           T$UBUS=·100000          WORD34=·000104          WORD63=·000176          WORD92=·000270
T$BTD·=·002000           T$1CLK=·000400          WORD35=·000106          WORD64=·000200          WORD93=·000272
T$CD·  =·000100          T$8BEN=·000020          WORD36=·000110          WORD65=·000202·         WORD94=·000274
T$CLK·=·002000           UBD.IN=·000020          WORD37=·000112·         WORD66=·000204          WORD95=·000276
T$DISK=·000200           WORD0  =·000000         WORD38=·000114          WORD67=·000206          WORD96=·000300
T$DRD·=·000004           WORD1  =·000002·        WORD39=·000116          WORD68=·000210          WORD97=·000302
T$EMEM=·010000           WORD10=·000024          WORD4  =·000010         WORD69=·000212·         WORD98=·000304
T$FSAA=·000000           WORD11=·000026          WORD40=·000120          WORD7  =·000016         WORD99=·000306
T$FSAB=·000004           WORD12=·000030          WORD41=·000122·         WORD70=·000214          WRDVAL=·000310
T$FSAC=·000014           WORD13=·000032·         WORD42=·000124          WORD71=·000216          XTREAD=·001000
T$FSB2=·000010           WORD14=·000034          WORD43=·000126          WORD72=·000220          XTWRTE=·000400
```

```
. ABS.   000000     000
         000000     001
SPPR·    000752·    002·
ERRORS·DETECTED:  0

VIRTUAL·MEMORY·USED:  3411 WORDS·  (·14 PAGES)
DYNAMIC·MEMORY:  4916 WORDS·  ( 18 PAGES)
ELAPSED·TIME:  00:00:46
>PPR,SPPR/-SP=[20,1]IM.[20,1]SPPR·
```

```
                                        .TITLE  SPREST
                                        .PSECT  SPREST
                                ;
                                ;
                                ;       HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
                                ;       SUBDOCUMENT PROCESSOR TEST ROUTINES
                                ;
                                ;       COMMANDS:
                                ;       PS      SWITCH QLB PAGES
                                ;
                                ;       ONCE A COMMAND HAS BBEN EXECUTED (OR AN ERROR ENCOUNTERED)
                                ;       THIS MODULE RETURNS CONTROL TO THE MODULE SP AT LOCATION
                                ;       'SPSXX'.
                                ;
000000                          PS5::
000000
000004  000167  000000G                CALL    KILL                    ;KILL AST
                                        JMP     SPSXX                   ;EXIT
                                ;
000010                          CL5::
000010
000014  000167  000000G                CALL    CL                      ;CALL ROUTINE IN MAIN
                                        JMP     SPSXX                   ;EXIT
                                ;
        000001                          .END
```

SYMBOL TABLE


```
ALUCKE= 040000          BYTE42= 000052          BYTE94= 000136          QR$LBR= 176424          Q$SM  = 100000
ALUOE = 004000          BYTE43= 000053          BYTE95= 000137          Q$ATTN= 000100          Q$SP  = 000120
A01   = 010000          BYTE44= 000054          BYTE96= 000140          Q$BCL = 000001          Q$SP2 = 000340
BITVAL= 000000          BYTE45= 000055          BYTE97= 000141          Q$CCCP= 000040          RGQ.EN= 000200
BIT0  = 000001          BYTE46= 000056          BYTE98= 000142          Q$CHB = 000400          RGQ.VA= 020000
BIT1  = 000002          BYTE47= 000057          BYTE99= 000143          Q$CHRL= 000200          SEQ.CI= 000010
BIT10 = 002000          BYTE48= 000060          BYTVAL= 000144          Q$CLR = 000040          SPSXX = ****** GX
BIT11 = 004000          BYTE49= 000061          CBKALL= 001000          Q$CNC = 030000          S$CLR = 000000
BIT12 = 010000          BYTE5 = 000005          CBKCLK= 000400          Q$CP  = 000060          S$LA  = 000001
BIT13 = 020000          BYTE50= 000062          CL    = ****** GX       Q$CPCC= 000010          S$QB  = 000005
BIT14 = 040000          BYTE51= 000063          CL5     000010RG  002   Q$CP2 = 000260          S$QR  = 000006
BIT15 = 100000          BYTE52= 000064          CNOBRE= 100000          Q$CSC = 010000          S$QX  = 000004
BIT2  = 000004          BYTE53= 000065          CPCCEN= 010000          Q$CSEL= 000360          S$SR  = 000007
BIT3  = 000010          BYTE54= 000066          CPREAD= 040000          Q$CSET= 000002          S$S1  = 000010
BIT4  = 000020          BYTE55= 000067          CPWRTE= 020000          Q$CSP = 020000          S$S2  = 000014
BIT5  = 000040          BYTE56= 000070          CSADRD= 000004          Q$DMA = 000001          TD$CTR= 176370
BIT6  = 000100          BYTE57= 000071          CSEQCI= 100000          Q$ENBK= 040000          TD$CTW= 176360
BIT7  = 000200          BYTE58= 000072          CSOE  = 000040          Q$ENOP= 020000          TD$INL= 004000
BIT8  = 000400          BYTE59= 000073          CSWRTE= 000100          Q$FAL = 004000          TD$MEM= 000270
BIT9  = 001000          BYTE6 = 000006          DBR.RD= 000001          Q$FC  = 000045          TD$OAR= 176344
BYTE0 = 000000          BYTE60= 000074          DB$CPP= 001457          Q$FD  = 000044          TD$OTR= 176346
BYTE1 = 000001          BYTE61= 000075          DB$SPT= 000026          Q$FP  = 000046          TD$QRD= 000274
BYTE10= 000012          BYTE62= 000076          DB$TPC= 000023          Q$HBF = 000002          TD$SW = 176376
BYTE11= 000013          BYTE63= 000077          DISPGS= 100000          Q$ICP = 000006          TD$TAR= 176372
BYTE12= 000014          BYTE64= 000100          DMAAWR= 000005          Q$IHB = 000003          TD$TAW= 176362
BYTE13= 000015          BYTE65= 000101          DMARRD= 000003          Q$IHRL= 000002          TD$TDR= 176374
BYTE14= 000016          BYTE66= 000102          DMARWR= 000004          Q$IMRP= 000007          TD$TDW= 176364
BYTE15= 000017          BYTE67= 000103          ENBR  = 010000          Q$LBD = 001000          T$AD  = 000020
BYTE16= 000020          BYTE68= 000104          KILL  = ****** GX       Q$LBDP= 001001          T$BA  = 000002
BYTE17= 000021          BYTE69= 000105          LOC.EN= 000100          Q$LBP = 000001          T$BD  = 000010
BYTE18= 000022          BYTE7 = 000007          LOC.WA= 040000          Q$LDCD= 000003          T$BSO = 100000
BYTE19= 000023          BYTE70= 000106          LOC.WB= 100000          Q$LDMD= 000004          T$BT  = 000020
BYTE2 = 000002          BYTE71= 000107          MAREN1= 000001          Q$LDPP= 002000          T$BTAR= 000030
BYTE20= 000024          BYTE72= 000110          MAREN2= 004000          Q$LHP = 010000          T$BTD = 002000
BYTE21= 000025          BYTE73= 000111          MARLOD= 010000          Q$MNC = 140000          T$CD  = 000100
BYTE22= 000026          BYTE74= 000112          MAROUT= 000002          Q$MR  = 000852          T$CLK = 002000
BYTE23= 000027          BYTE75= 000113          MAR.LO= 002000          Q$MRP = 000040          T$DISK= 000200
BYTE24= 000030          BYTE76= 000114          MAR.OU= 000040          Q$MRP2= 000240          T$DRD = 000004
BYTE25= 000031          BYTE77= 000115          MBKALL= 001000          Q$MSC = 040000          T$EMEM= 010000
BYTE26= 000032          BYTE78= 000116          MBKCLK= 000400          Q$MSET= 000004          T$FSAA= 000000
BYTE27= 000033          BYTE79= 000117          MMADRD= 000100          Q$MSP = 100000          T$FSAB= 000004
BYTE28= 000034          BYTE8 = 000010          MMLEFT= 000002          Q$NCLK= 176000          T$FSAC= 000014
BYTE29= 000035          BYTE80= 000120          MMOE  = 000004          Q$PP  = 000100          T$FSB2= 000010
BYTE3 = 000003          BYTE81= 000121          MMWRTE= 000010          Q$PPSW= 000320          T$IB  = 000026
BYTE30= 000036          BYTE82= 000122          MNOBRE= 100000          Q$PP2 = 000300          T$IBAR= 000024
BYTE31= 000037          BYTE83= 000123          MREN1 = 000001          Q$QHLT= 000013          T$IBE = 020000
BYTE32= 000040          BYTE84= 000124          MREN2 = 020000          Q$QL  = 000043          T$IBF = 040000
BYTE33= 000041          BYTE85= 000125          MSYN  = 000040          Q$QLA = 000053          T$ICD = 000040
BYTE34= 000042          BYTE86= 000126          N     = 000144          Q$QLB = 000054          T$MODE= 004000
BYTE35= 000043          BYTE87= 000127          PLB   = 000010          Q$QLR = 000001          T$OB  = 000036
BYTE36= 000044          BYTE88= 000130          PLC   = 000020          Q$QW  = 000042          T$OBE = 004000
BYTE37= 000045          BYTE89= 000131          PLD   = 000030          Q$RDCD= 000005          T$OBF = 010000
BYTE38= 000046          BYTE9 = 000011          PLRWR = 000200          Q$RDMD= 000006          T$OBRA= 000034
BYTE39= 000047          BYTE90= 000132          PLR.EN= 000200          Q$REBK= 001000          T$OBWA= 000032
BYTE4 = 000004          BYTE91= 000133          PS5     000000RG  002   Q$RNC = 006000          T$OUTA= 100000
BYTE40= 000050          BYTE92= 000134          QR$CR1= 176420          Q$RSC = 004000          T$RBDO= 000200
BYTE41= 000051          BYTE93= 000135          QR$CR2= 176422          Q$RSET= 000010          T$RNB = 000040
```

```
SPREST· M[illegible]0·M1110  27-MAR-80 15:33  PAGE·5-2·
SYMBOL·TA[illegible]·
```

```
T$RSET=·040000          WORD2·=·000004          WORD40=·000120          WORD61=·000172·         WORD82=·000244
T$SC··=·000022·         WORD20=·000050          WORD41=·000122·         WORD62=·000174          WORD83=·000246
T$SCLK=·020000          WORD21=·000052·         WORD42=·000124          WORD63=·000176          WORD84=·000250
T$SEG1=·000000          WORD22=·000054          WORD43=·000126          WORD64=·000200          WORD85=·000252
T$SEG2=·000001          WORD23=·000056          WORD44=·000130          WORD65=·000202·         WORD86=·000254
T$SEG3=·000002·         WORD24=·000060          WORD45=·000132·         WORD66=·000204          WORD87=·000256
T$SO··=·000001          WORD25=·000062·         WORD46=·000134          WORD67=·000206          WORD88=·000260
T$UBUS=·100000          WORD26=·000064          WORD47=·000136          WORD68=·000210          WORD89=·000262
T$1CLK=·000400          WORD27=·000066          WORD48=·000140          WORD69=·000212·         WORD9 =·000022
T$8BEN=·000020          WORD28=·000070          WORD49=·000142·         WORD7 =·000016          WORD90=·000264
UBD.IN=·000020          WORD29=·000072·         WORD5 =·000012·         WORD70=·000214          WORD91=·000266
WORD0 =·000000          WORD3 =·000006          WORD50=·000144          WORD71=·000216          WORD92=·000270
WORD1 =·000002·         WORD30=·000074          WORD51=·000146          WORD72=·000220          WORD93=·000272
WORD10=·000024          WORD31=·000076          WORD52=·000150          WORD73=·000222·         WORD94=·000274
WORD11=·000026          WORD32=·000100          WORD53=·000152·         WORD74=·000224          WORD95=·000276
WORD12=·000030          WORD33=·000102·         WORD54=·000154          WORD75=·000226          WORD96=·000300
WORD13=·000032·         WORD34=·000104          WORD55=·000156          WORD76=·000230          WORD97=·000302
WORD14=·000034          WORD35=·000106          WORD56=·000160          WORD77=·000232·         WORD98=·000304
WORD15=·000036          WORD36=·000110          WORD57=·000162·         WORD78=·000234          WORD99=·000306
WORD16=·000040          WORD37=·000112·         WORD58=·000164          WORD79=·000236          WRDVAL=·000310
WORD17=·000042·         WORD38=·000114          WORD59=·000166          WORD8 =·000020          XTREAD=·001000
WORD18=·000044          WORD39=·000116          WORD6 =·000014          WORD80=·000240          XTWRTE=·000400
WORD19=·000046          WORD4 =·000010          WORD60=·000170          WORD81=·000242·
```

```
. ABS.  000000     000
        000000     001
SPREST· 000020     002·
ERRORS·DETECTED:  0

VIRTUAL·MEMORY·USED:  3001 WORDS·  (·12·PAGES)
DYNAMIC·MEMORY:  3860 WORDS·  ( 14 PAGES)
ELAPSED·TIME:  00:00:39
SPREST,SPREST/-SP=[20,1]IM,[20,1]SPREST·
```

```
                                        .TITLE  $PSUB
000000                                  .PSECT  $PSUB
                                ;
                                        .MCALL  WTSE$S,CLEF$S
                                ;
          000003                EFN.3   =       3
                                ;
                                ;
                                ;       HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
                                ;       SUBDOCUMENT PROCESSOR SUBROUTINES
                                ;
                                ;
                                ;       DATA TRANSFER
                                ;       LOD BUS REGISTER TO A DESTINATION ON THE SP BUS
                                ;
                                ;       INPUT:
                                ;       2(SP)   DATA FOR PRE-SELECTED SP DESTINATION
                                ;
                                ;
000000                          LBSP::
000000  016667  000002  176424          MOV     2(SP),@R$LBR            ;MOVE DATA TO LOD BUS REG
000006  012746  001001                  MOV     #<Q$LBD+Q$LBP>,-(SP)    ;CLR DRIVE AND PULSE
000012  052716  000360                  BIS     #<Q$CSEL>,(SP)          ;CLR SELECTION BITS
000016  012746  176000                  MOV     #<Q$NCLK>,-(SP)         ;SET NO-CLOCKS
000022  052716  000340                  BIS     #Q$SP2,(SP)             ;SELECT SP
000026                                  CALL    CSR1                    ;
                                ;
000032  012746  006000                  MOV     #Q$RNC,-(SP)            ;CLEAR CP NO-CLOCK BITS
000036  012746  001000                  MOV     #Q$LBD,-(SP)            ;SET LOD BUS DRIVE
000042                                  CALL    CSR1                    ;
                                ;
                                ;       DE-SELECTION
                                ;
000046  012746  001001                  MOV     #<Q$LBD+Q$LBP>,-(SP)    ;CLEAR DRIVE AND PULSE
000052  052716  000360                  BIS     #<Q$CSEL>,(SP)          ;CLR SELECTION BITS
000056  012746  176000                  MOV     #<Q$NCLK>,-(SP)         ;SET NO-CLOCKS
000062                                  CALL    CSR1                    ;
                                ;
000066  011666  000002                  MOV     (SP),2(SP)              ;MOVE RETURN ADDRESS DOWN STACK
000072  005726                          TST     (SP)+                   ;POINT TO RETURN ADDRESS
000074                                  RETURN
```

```
                                        ;
                                        ;
                                        ;       DATA·TRANSFER·
                                        ;       LOD·BUS·REGISTER·TO·A·DESTINATION·ON·THE·SP·BUS·
                                        ;       SINGLE·CLOCK SEQUENCER ONLY·
                                        ;
                                        ;       INPUT:
                                        ;       2(SP)    DATA FOR·PRE-SELECTED·SP DESTINATION·
                                        ;
                                        ;
000076                                  LBSSC::
000076  016667  000002  176424                  MOV·    2(SP),@R$LBR·           ;MOVE·DATA TO·LOD·BUS·REG·
000104  012746  001001                          MOV·    #<Q$LBD+Q$LBP>,-(SP)    ;CLEAR·DRIVE·AND·PULSE·
000110  052716  000360                          BIS·    #Q$CSEL,(SP)            ;CLR·SELECTION·BITS·
000114  012746  176000                          MOV·    #<Q$NCLK>,-(SP)         ;SET·NO-CLOCKS·
000120  052716  000340                          BIS·    #Q$SP2,(SP)             ;SELECT·SP
000124                                          CALL·   CSR1                    ;WRITE·CONTROL·REGISTER·
                                        ;
000130  012746  006000                          MOV·    #Q$RNC,-(SP)            ;CLEAR·SP·NO-CLOCK·BITS·
000134  012746  005000                          MOV·    #<Q$RSC+Q$LBD>,-(SP)    ;SET·SP·CLOCK·
000140                                          CALL·   CSR1                    ;
                                        ;
                                        ;       DE-SELECTION
                                        ;
000144  012746  001001                          MOV·    #<Q$LBD+Q$LBP>,-(SP)    ;CLEAR·DRIVE·AND·PULSE·
000150  052716  000360                          BIS·    #Q$CSEL,(SP)            ;CLR·SELECTION·BITS·
000154  012746  176000                          MOV·    #<Q$NCLK>,-(SP)         ;SET·NO·CLOCKS·
000160                                          CALL·   CSR1
                                        ;
000164  011666  000002                          MOV·    (SP),2(SP)              ;MOVE·RETURN·ADDRESS DOWN·STACK·
000170  005726                                  TST·    (SP)+                   ;POINT·TO·RETURN·ADDRESS·
000172                                          RETURN·
```

```
                                        ;
                                        ;
                                        ;       DATA·TRANSFER·TO·LOD·BUS·REG·FROM·SP·
                                        ;
                                        ;       OUTPUT:
                                        ;       (SP)    DATA FROM·PRE-SELECTED·SP·SOURCE·
                                        ;
                                        ;
000174                                  SPLB::
000174  012746  001001                          MOV·    #<Q$LBD+Q$LBP>,-(SP)    ;CLEAR·DRIVE·AND·PULSE·
000200  052716  176360                          BIS·    #<Q$CSEL+Q$NCLK>,(SP)   ;CLR·SELECTION·BITS/NO·CLOCKS·
000204  012746  000340                          MOV·    #<Q$SP2>,-(SP)          ;SET·NO-CLOCKS·
000210                                          CALL·   CSR1                    ;
000214  011646                                  MOV·    (SP),-(SP)              ;MOVE·RETURN·ADDR·UP STACK·
000216  016766  176424  000002·                 MOV·    QR$LBR,2(SP)            ;MOVE·DATA ONTO·STACK·
                                        ;
000224  012746  000340                          MOV·    #Q$SP2,-(SP)            ;CLEAR·SP·SELECT·
000230  012746  176000                          MOV·    #<Q$NCLK>,-(SP)         ;SET·NO-CLOCKS·
000234                                          CALL·   CSR1                    ;
000240                                          RETURN·
```

```
 97                                     ;
 98                                     ;
 99                                     ;       SP CONTROL REGISTER LOADING
100                                     ;
101                                     ;       INPUT:
102                                     ;       2(SP)   BIT SETTING FOR SP CONTROL REGISTER
103                                     ;
104                                     ;
105 000242                              SPCR::
106 000242  016667  000002  176424              MOV     2(SP),QR$LBR            ;CONTROL BITS DESTINED FOR SP
107 000250  012746  001001                      MOV     #<Q$LBD+Q$LBP>,-(SP)    ;CLEAR DRIVE AND PULSE
108 000254  052716  000360                      BIS     #Q$CSEL,(SP)            ;CLR SELECTION BITS
109 000260  012746  000120                      MOV     #Q$SP,-(SP)             ;SELECT SP
110 000264                                      CALL    CSR1                    ;
111                                     ;
112 000270  005046                              CLR     -(SP)                   ;CLEAR NOTHING
113 000272  012746  000001                      MOV     #Q$LBP,-(SP)            ;SET PULSE
114 000276                                      CALL    CSR1                    ;
115                                     ;
116 000302  012746  000121                      MOV     #<Q$SP+Q$LBP>,-(SP)     ;CLEAR CR SELECTION AND PULSE
117 000306  005046                              CLR     -(SP)                   ;SET NOTHING
118 000310                                      CALL    CSR1
119                                     ;
120 000314  011666  000002                      MOV     (SP),2(SP)              ;MOVE RETURN ADDRESS DOWN STACK
121 000320  005726                              TST     (SP)+                   ;POINT TO RETURN ADDRESS
122 000322                                      RETURN
123                                     ;
124         000001                              .END
```

SPSUB M[illegible]RO M1110 27-MAR-80 15:34 PAGE 8-1
SYMBOL TA[illegible]

```
ALUCKE= 040000      BYTE42= 000052      BYTE94= 000136      QR$LBR= 176424      Q$SM  = 100000
ALUOE = 004000      BYTE43= 000053      BYTE95= 000137      Q$ATTN= 000100      Q$SP  = 000120
A01   = 010000      BYTE44= 000054      BYTE96= 000140      Q$BCL = 000001      Q$SP2 = 000340
BITVAL= 000000      BYTE45= 000055      BYTE97= 000141      Q$CCCP= 000040      RGQ.EN= 000200
BIT0  = 000001      BYTE46= 000056      BYTE98= 000142      Q$CHB = 000400      RGQ.VA= 020000
BIT1  = 000002      BYTE47= 000057      BYTE99= 000143      Q$CHRL= 000200      SEQ.CI= 000010
BIT10 = 002000      BYTE48= 000060      BYTVAL= 000144      Q$CLR = 000040      SPCR    000242RG    002
BIT11 = 004000      BYTE49= 000061      CBKALL= 001000      Q$CNC = 030000      SPLB    000174RG    002
BIT12 = 010000      BYTE5 = 000005      CBKCLK= 000400      Q$CP  = 000060      S$CLR = 000000
BIT13 = 020000      BYTE50= 000062      CNOBRE= 100000      Q$CPCC= 000010      S$LA  = 000001
BIT14 = 040000      BYTE51= 000063      CPCCEN= 010000      Q$CP2 = 000260      S$QB  = 000005
BIT15 = 100000      BYTE52= 000064      CPREAD= 040000      Q$CSC = 010000      S$QR  = 000006
BIT2  = 000004      BYTE53= 000065      CPWRTE= 020000      Q$CSEL= 000360      S$QX  = 000004
BIT3  = 000010      BYTE54= 000066      CSADRD= 000004      Q$CSET= 000002      S$SR  = 000007
BIT4  = 000020      BYTE55= 000067      CSEQCI= 100000      Q$CSP = 020000      S$S1  = 000010
BIT5  = 000040      BYTE56= 000070      CSOE  = 000040      Q$DMA = 000001      S$S2  = 000014
BIT6  = 000100      BYTE57= 000071      CSR1  = ****** GX   Q$ENBK= 040000      TD$CTR= 176370
BIT7  = 000200      BYTE58= 000072      CSWRTE= 000100      Q$ENOP= 020000      TD$CTU= 176360
BIT8  = [illegible]
BIT9  = [illegible]
[illegible]
[illegible]
```

```
BYTE12=·000014      BYTE64=·000100      DMARRD=·000003      Q$IHB =·000002      T$BTAR=·176372·
BYTE13=·000015          BYTE65=·000101      DMAWRT=·000004      Q$IMRL=·000002·      TD$TAW=·176362
BYTE14=·000016          BYTE66=·000102·     EFN.3 =·000003      Q$IMRP=·000007      TD$TDR=·176374
BYTE15=·000017          BYTE67=·000103      ENBR· =·010000      Q$LBD·=·001000      TD$TDW=·176364
BYTE16=·000020          BYTE68=·000104      LBSP·   000000RG·   002·Q$LBDP=·001001      T$AD  =·000020
BYTE17=·000021          BYTE69=·000105      LBSSC·  000076RG·   002·Q$LBP =·000001      T$BA  =·000002
BYTE18=·000022·         BYTE7 =·000007      LOC.EN=·000100      Q$LDCD=·000003      T$BD  =·000010
BYTE19=·000023          BYTE70=·000106      LOC.WA=·040000      Q$LDMD=·000004      T$BSO·=·100000
BYTE2·=·000002·         BYTE71=·000107      LOC.WB=·100000      Q$LDPP=·002000      T$BT  =·000020
BYTE20=·000024          BYTE72=·000110      MAREN1=·000001      Q$LHP =·010000      T$BTAR=·000030
BYTE21=·000025          BYTE73=·000111      MAREN2=·004000      Q$MNC·=·140000      T$BTD·=·002000
BYTE22=·000026          BYTE74=·000112·     MARLOD=·010000      Q$MR·=·000052·      T$CD  =·000100
BYTE23=·000027          BYTE75=·000113      MAROUT=·000002·     Q$MRP =·000040      T$CLK·=·002000
BYTE24=·000030          BYTE76=·000114      MAR.LO=·002000      Q$MRP2=·000240      T$DISK=·000200
BYTE25=·000031          BYTE77=·000115      MAR.OU=·000040      Q$MSC·=·040000      T$DRD·=·000004
BYTE26=·000032·         BYTE78=·000116      MBKALL=·001000      Q$MSET=·000004      T$EMEM=·010000
BYTE27=·000033          BYTE79=·000117      MBKCLK=·000400      Q$MSP =·100000      T$FSAA=·000000
BYTE28=·000034          BYTE8 =·000010      MMADRD=·000100      Q$NCLK=·176000      T$FSAB=·000004
BYTE29=·000035          BYTE80=·000120      MMLEFT=·000002·     Q$PP·=·000100      T$FSAC=·000014
BYTE3 =·000003          BYTE81=·000121      MMOE· =·000004      Q$PPSW=·000320      T$FSB2=·000010
BYTE30=·000036          BYTE82=·000122·     MMWRTE=·000010      Q$PP2·=·000300      T$IB  =·000026
BYTE31=·000037          BYTE83=·000123      MNOBRE=·100000      Q$QHLT=·000013      T$IBAR=·000024
BYTE32=·000040          BYTE84=·000124      MREN1 =·000001      Q$QL·=·000043      T$IBE·=·020000
BYTE33=·000041          BYTE85=·000125      MREN2·=·020000      Q$QLA·=·000053      T$IBF·=·040000
BYTE34=·000042·         BYTE86=·000126      MSYN·=·000040      Q$QLB·=·000054      T$ICD·=·000040
BYTE35=·000043          BYTE87=·000127      N·    =·000144      Q$QLR·=·000001      T$MODE=·004000
BYTE36=·000044          BYTE88=·000130      PLB·  =·000010      Q$QW·=·000042·      T$OB  =·000036
BYTE37=·000045          BYTE89=·000131      PLC·  =·000020      Q$RDCD=·000005      T$OBE·=·004000
BYTE38=·000046          BYTE9 =·000011      PLD·  =·000030      Q$RDMD=·000006      T$OBF·=·010000
BYTE39=·000047          BYTE90=·000132·     PLRWR·=·000200      Q$REBK=·001000      T$OBRA=·000034
BYTE4 =·000004          BYTE91=·000133      PLR.EN=·000200      Q$RNC·=·006000      T$OBWA=·000032
BYTE40=·000050          BYTE92=·000134      QR$CR1=·176420      Q$RSC·=·004000      T$OUTA=·100000
BYTE41=·000051          BYTE93=·000135      QR$CR2=·176422·     Q$RSET=·000010      T$RBDO=·000200
```

```
SPSUB···M[illegible]0·M1110  27-MAR-80 15:34  PAGE·8-2·
SYMBOL·TA[illegible]·

T$RNB·=·000040          WORD19=·000046          WORD4 =·000010          WORD60=·000170          WORD81=·000242
T$RSET=·040000          WORD2·=·000004          WORD40=·000120          WORD61=·000172·         WORD82=·000244
T$SC··=·000022·         WORD20=·000050          WORD41=·000122·         WORD62=·000174          WORD83=·000246
T$SCLK=·020000          WORD21=·000052·         WORD42=·000124          WORD63=·000176          WORD84=·000250
T$SEG1=·000000          WORD22=·000054          WORD43=·000126          WORD64=·000200          WORD85=·000252
T$SEG2=·000001          WORD23=·000056          WORD44=·000130          WORD65=·000202·         WORD86=·000254
T$SEG3=·000002·         WORD24=·000060          WORD45=·000132·         WORD66=·000204          WORD87=·000256
T$SO··=·000001          WORD25=·000062·         WORD46=·000134          WORD67=·000206          WORD88=·000260
T$UBUS=·100000          WORD26=·000064          WORD47=·000136          WORD68=·000210          WORD89=·000262
T$1CLK=·000400          WORD27=·000066          WORD48=·000140          WORD69=·000212·         WORD9 =·000022
T$8BEN=·000020          WORD28=·000070          WORD49=·000142·         WORD7 =·000016          WORD90=·000264
UBD.IN=·000020          WORD29=·000072·         WORD5 =·000012·         WORD70=·000214          WORD91=·000266
WORD0 =·000000          WORD3 =·000006          WORD50=·000144          WORD71=·000216          WORD92=·000270
WORD1 =·000002·         WORD30=·000074          WORD51=·000146          WORD72=·000220          WORD93=·000272
WORD10=·000024          WORD31=·000076          WORD52=·000150          WORD73=·000222·         WORD94=·000274
WORD11=·000026          WORD32=·000100          WORD53=·000152·         WORD74=·000224          WORD95=·000276
WORD12=·000030          WORD33=·000102·         WORD54=·000154          WORD75=·000226          WORD96=·000300
WORD13=·000032·         WORD34=·000104          WORD55=·000156          WORD76=·000230          WORD97=·000302
WORD14=·000034          WORD35=·000106          WORD56=·000160          WORD77=·000232·         WORD98=·000304
WORD15=·000036          WORD36=·000110          WORD57=·000162·         WORD78=·000234          WORD99=·000306
WORD16=·000040          WORD37=·000112·         WORD58=·000164          WORD79=·000236          WRDVAL=·000310
WORD17=·000042·         WORD38=·000114          WORD59=·000166          WORD8 =·000020          XTREAD=·001000
WORD18=·000044          WORD39=·000116          WORD6 =·000014          WORD80=·000240          XTWRTE=·000400

. ABS.  000000     000
        000000     001
SPSUB·  000324     002·
ERRORS·DETECTED:  0

VIRTUAL MEMORY USED:  3112·WORDS·  (·13 PAGES)
DYNAMIC·MEMORY:  3860 WORDS·  ( 14 PAGES)
ELAPSED·TIME:  00:00:42·
>PSUB,SPSUB/-SP=[20,1]IM,[20,1]SPSUB·
```

```
TASK·   NAME·   : AIDQR·
PARTITION·NAME  : GEN
IDENTIFICATION  : 08
TASK· UIC·      : [20,3]·
STACK·   LIMITS: 040176 041175 001000 00512.
PRG·XFR·ADDRESS: 043372·
TASK·ATTRIBUTES: AL,CP·
TOTAL·ADDRESS·WINDOWS: 1.
TASK· IMAGE· SIZE·  : 7744. WORDS·
TASK·ADDRESS·LIMITS: 040000 076167
R-W DISK BLK·LIMITS: 000042·000137 000076 00062.


AIDQR.TSK;7 OVERLAY·DESCRIPTION:

BASE·   TOP·      LENGTH·
----    ---       ------
040000  072013  032014  13324.      QMAIN·
072014  072363  000350  00232.           MRP·
072364  073557  001174  00636.                MRLD·
072364  073473  001110  00584.                MRPR·
072364  073657  001274  00700.                MRREST·
072364  073353  000770  00504.                MRBUG·
072014  073517  001504  00836.           CP·
073520  074373  000654  00428.                CPBUG1
073520  076167  002450  01320.                CPBUG2·
073520  075327  001610  00904.                CPREST·
073520  075407  001670  00952.                CPLD·
073520  075303  001564  00884.                CPPR·
072014  072647  000634  00412.           BCE·
072014  072567  000554  00364.           PPS·
072570  075233  002444  01316.                PPLD·
072570  075167  002400  01280.                PPPR·
072570  074167  001400  00768.                PPREST·
072014  072573  000560  00368.           SP·
072574  073647  001054  00556.                SPLD·
072574  073547  000754  00492.                SPPR·
072574  07261.  000020  00016.                SPREST
```

```
AIDQR.TSK;7    MEMORY ALLOCATION MAP  TKB                PAGE 2
QMAIN·                27-MAR-80

*** ROOT·SEGMENT: QMAIN·

R/W·MEM· LIMITS: 040000 072013 032014 13324.
DISK·BLK·LIMITS: 000042·000074 000033 00027.

MEMORY·ALLOCATION·SYNOPSIS:

SECTION,·                                          TITLE,·· IDENT, FILE,··
--------                                           -----    -----  ----
. BLK,:(RW,I,LCL,REL,CON) 041176 011660 05040.
                          041176 005116 02638. QMAIN·          QMAIN.OBJ;1
CPSUB·:(RW,I,LCL,REL,CON) 053056 000502·00322.
                          053056 000502·00322. CPSUB·          CPSUB.OBJ;1
MRPSUB:(RW,I,LCL,REL,CON) 053560 000506 00326.
                          053560 000506 00326. MRPSUB·         MRPSUB.OBJ;1
PPSUB·:(RW,I,LCL,REL,CON) 054266 000766 00502.
                          054266 000766 00502. PPSUB·          PPSUB.OBJ;1
$$ALER:(RW,I,LCL,REL,CON) 055254 000024 00020.
$$ALVC:(RW,D,LCL,REL,CON) 055300 000060 00048.
$$AUTO:(RW,I,LCL,REL,CON) 055360 000130 00088.
$$FSR1:(RW,D,GBL,REL,OVR) 055510 001020 00528.
                          055510 001020 00528. QMAIN·          QMAIN.OBJ;1
$$FSR2:(RW,D,GBL,REL,CON) 056530 000104 00068.
$$MRKS:(RO,I,LCL,REL,OVR) 071544 000076 00062.
$$OVDT:(RW,D,LCL,REL,OVR) 056634 000020 00016.
$$OVRS:(RW,I,LCL,ABS,CON) 000000 000000 00000.
$$RDSG:(RO,I,LCL,REL,OVR) 071642·000150 00104.
$$RESL:(RW,I,LCL,REL,CON) 056654 012270 05304.
$$RGDS:(RW,D,LCL,REL,CON) 071144 000000 00000.
$$RTS·:(RW,I,GBL,REL,OVR) 071144 000002·00002.
$$SGD0:(RW,D,LCL,REL,OVR) 071146 000000 00000.
$$SGD1:(RW,D,LCL,REL,CON) 071146 000374 00252.
$$SGD2:(RW,D,LCL,REL,OVR) 071542·000002·00002.
$$WNDS:(RW,D,LCL,REL,CON) 071544 000000 00000.

GLOBAL·SYMBOLS:

APLACE 041234-R·  BUFS3  043754-R·  CSR1   043606-R·  ENFILE·046026-R·  ERWORD 041226-R·  HLLOW  041452-R  LHLOW  041446-R·
ASCIZ  043063-R·  BUFS4  043736-R·  DATA1  041404-R·  ERR1   046122-R·  FAHIGH·041440-R·  INCVAL·041376-R  LOOP   000004
ASTFLG 000200     CDHIGH·041430-R·  DATA2  041406-R·  ERR10  046056-R·  FALOW  041442-R·  IO.DET 002000     LOOPR· 044602-R·
ASTWRD 041364-R·  CDLOW  041432-R·  DATA3  041410-R·  ERR11  046052-R·  FIND   045206-R·  IO.RVB 010400     LUN.TT 000001
BASE·  041232-R·  CL     043646-R·  DATA4  041412-R·  ERR12· 046046-R·  GCMBLK·043064-R·  IO.WVB·011000     MDHIGH·041420-R·
3CE·   055320-R·  COMXX· 043532-R·  EFBUF· 041212-R·  ERR2·  046116-R·  GCMBUF·041236-R·  KILL·  044762-R·  MDLOW· 041422-R·
JINWD· 041230-R·  CONSOL 045634-R·  EFN.1  000001     ERR3   046112-R·  GCMLEN·041360-R·  LBCP·  053224-R  MEND·  041374-R·
3LHIGH·041454-R·  CP·    055310-R·  EFN.2· 000002·    ERR4   046106-R·  GCMPNT·041362-R·  LBCSC  053322-R  MMHIGH·041414-R·
3LLOW· 041456-R·  CPCR·  053470-R·  EFN.3  000003     ERR5   046102-R·  GCONLY·044452-R·  LBMRP  053732-R  MMLOW  041416-R·
BREAK· 004000     CPCRA· 053476-R·  EFN.33 000041     ERR6   046076-R·  G.DPRM·000160     LBMSC  054030-R  MRP    055300-R·
BUFSET·044010-R·  CPLB·  053420-R·  EFN.4  000004     ERR7   046072-R·  HANG·  044624-R·  LBPP·  054266-R  MRPCR· 054176-R·
BUFSM· 044026-R·  CSHIGH·041424-R·  ENDMEM 046032-R·  ERR8   046066-R·  HANG2· 044710-R·  LBPSC  054364-R  MRPCRA·054204-R·
BUFS2· 043772-R·  CSLOW· 041426-R·  ENDTST 046016-R·  ERR9   046062-R·  HLHIGH·041450-R·  LHHIGH·041444-R  MRPLB· 054126-R·
```

```
MSTRT  041370-R  OUT1    046042-R  PRINT  041746-R  SDHIGH 041470-R  SQHIGH 041460-R  TRTBL   041474-R  $CEFI  005174
MSTR2  041372-R  PACK    045324-R  QXHIGH 041434-R  SDLOW  041472-R  SQLOW  041462-R  TRTBL2  041673-R  $DIV   007146
MYSELF 041176-R  PDATA   044162-R  QXLOW  041436-R  SELPG  054614-R  SRHIGH 041464-R  TSKTCB  041202-R  $DRDSE 017134
NEQLB  010000    PPCR    054532-R  RP     001000    SEQCS  053056-R  SRLOW  041466-R  UNPK    045544-R  $MUL   007116
OLDVEC 041204-R  PPLB    054462-R  RSPCNT 041400-R  SEQMM  053560-R  STAT   041222-R  UPLIM   041402-R  $TKTCB 004026
ONCE   000100    PPS     055330-R  RTNPT  041366-R  SPCR   055350-R  STOP   046036-R  WRTCS   053144-R
OUT    002000    PRDATA  044354-R  SCAN   045114-R  SPS    055340-R  TIME   046022-R  WRTMM   053646-R
```

*** SEGMENT: MRP

R/W MEM LIMITS: 072014 072363 000350 00232.
DISK BLK LIMITS: 000075 000075 000001 00001.

MEMORY ALLOCATION SYNOPSIS:

SECTION				TITLE	IDENT	FILE
. BLK:	(RW,I,LCL,REL,CON)	072014	000000	00000.		
MRP	(RW,I,LCL,REL,CON)	072014	000226	00150.		
		072014	000226	00150.	MRP	MRP;OBJ;1
$$ALVC:	(RW,D,LCL,REL,CON)	072242	000120	00080.		
$$RTS	(RW,I,GBL,REL,OVR)	071144	000002	00002.		

GLOBAL SYMBOLS:

AT1	072322-R	GO1	072332-R	MRP	072064-R	OF1	072342-R	RE1	072272-R	SS1	072352-R				
CL1	072262-R	LD1	072242-R	MRPXX	072144-R	PR1	072252-R	RS1	072302-R	ST1	072312-R				