```
000001                          TITLE MTUS3,'REV G'  NINE TRACK PE/NRZI MAG TAPE T&V (SAF)
000002                  *
000003                  *
000004                  *
000005                  *
000006                  * DESCRIPTION:
000007                  *
000008                  *    THIS T&V PROGRAM VERIFIES PROPER OPERATION OF A LEVEL 6 NINE TRACK
000009                  *    MAGNETIC TAPE SUBSYSTEM. IT PROVIDES A FIRST LEVEL OF DIAGNOSIS
000010                  *    WHEN FAILURES ARE DETECTED, AND MAKES A SET OF FACILITIES AVAILABLE
000011                  *    TO AID IN MORE EXTENSIVE PROBLEM INVESTIGATIONS.
000012                  *
000013                  *    THE SUBSYSTEM ITEMS SUPPORTED BY THIS PROGRAM ARE:
000014                  *
000015                  *       MTC9101  MAG TAPE CONTROLLER
000016                  *       MTM9102  9 TRACK NRZI PAC (ADAPTER)
000017                  *       MTC9102  9 TRACK CONTROLLER AND ADAPTER
000018                  *
000019                  *       MTU9104  9 TRACK 45 IPS 800 BPI NRZI DRIVE
000020                  *       MTU9105  9 TRACK 75 IPS 800 BPI NRZI DRIVE
000021                  *       MTU9109  9 TRACK 45 IPS 800 BPI NRZI/1600 BPI PE DRIVE
000022                  *       MTU9110  9 TRACK 75 IPS 800 BPI NRZI/1600 BPI PE DRIVE
000023                  *       MTU9114  9 TRACK 45 IPS 1600 BPI PE DRIVE
000024                  *       MTU9115  9 TRACK 75 IPS 1600 BPI PE DRIVE
000025                  *
000026                  *
000027                  * REVISION HISTORY:
000028                  *
000029                  *       REV     DATE
000030                  *        A      SEPT 76     ORIGINAL RELEASE
000031                  *        B      DEC  76
000032                  *        C      FEB  77
000033                  *        D      JUNE 77     SAF MODE ADDED
000034                  *        E      JULY 77     MLCP LIBRARY ADDED
000035                  *        F      APR  78     PE TAPE ADDED
000036                  *        G      JUNE 78
000037                  *
000038                  *
000039                  ********************************************************************
000040                  *
000041                  *    THIS  DOCUMENT  AND  THE  INFORMATION  CONTAINED
000042                  *    THEREIN IS CONFIDENTIAL AND PROPRIETARY TO AND THE
000043                  *    EXCLUSIVE  PROPERTY  OF  HONEYWELL INFORMATION
000044                  *    SYSTEMS INC.  IT  IS MADE AVAILABLE ONLY TO HONEY-
000045                  *    WELL AUTHORIZED RECIPIENTS FOR THEIR USE SOLELY IN
000046                  *    THE  MAINTENANCE  AND  OPERATION  OF  HONEYWELL
000047                  *    PRODUCTS.  THIS  DOCUMENT  AND INFORMATION MUST BE
000048                  *    MAINTAINED IN STRICTEST CONFIDENCE;  IT  MUST  NOT
000049                  *    BE  REPRODUCED  IN  WHOLE OR IN PART; AND IT SHALL
000050                  *    NOT BE DISCLOSED TO ANY OTHER  PARTY  WITHOUT  THE
000051                  *    PRIOR WRITTEN CONSENT OF HONEYWELL.
000052                  *
000053                  ********************************************************************
```

```
000054                  /*****************************************************************
000055                  *
000056                  * PROGRAM PREPARATION:
000057                  *
000058                  *    THE ROOT SOURCE OF THIS PROGRAM, AFTER THE ADDITION OF APPROPRIATE
000059                  *    TITLE AND END STATEMENTS, WAS PROCESSED BY THE HOST RESIDENT ASSEMBLER
000060                  *    TO CREATE EITHER SHORT OR LONG ADDRESS FORM (SAF OR LAF) OBJECT TEXT
000061                  *    AND LISTING. THE OBJECT TEXT WAS FURTHER PROCESSED BY THE HOST RESIDENT
000062                  *    LINKER USING THE APPROPRIATE CONSOLE ZV$LIB LIBRARY TO CREATE A PUNCH
000063                  *    SEGMENT CONTAINING AN EXECUTABLE MODULE. THE ASSEMBLY LISTING WAS
000064                  *    AUGMENTED WITH CROSS REFERENCE DATA PLUS THE LOAD MAP FROM THE LINKER
000065                  *    TO CREATE A LIST SEGMENT.
000066                  *
000067                  *                ROOT            SAF             LAF
000068                  *                ----            ---             ---
000069                  *        NAME    MTUX3           MTUS3           MTUL3
000070                  *        DOCUMENT 60133890-007   60131897-007   60133889-007
000071                  *
000072                  *
000073                  * PROGRAM DISTRIBUTION:
000074                  *
000075                  *    THE ELEMENTARY ITEMS SUBMITTED TO THE I&V PROGRAM DISTRIBUTION CENTER
000076                  *    WERE THE EXECUTABLE LINKED IMAGES OF MTUS3 AND MTUL3 ON DISKETTE,
000077                  *    AND MAGNETIC TAPE IMAGES OF THE AUGMENTED LISTINGS.
000078                  *
000079                  *    REPRODUCTIONS OF THE EXECUTABLE LINKED IMAGES MAY BE AS CARD DECKS
000080                  *    OR AS A MEMBER OF A MULTIPLE MEMBER FILE. IN THE MOST FREQUENT CASE
000081                  *    THEY WILL BE FOUND AS MEMBER "SM" (SAF) OR "LM" (LAF) WITHIN FILE
000082                  *    "PROGFILE" OF A DISKETTE VOLUME ENTITLED "DIAGS".
000083                  *
000084                  *    DISTRIBUTION OF THE LISTINGS, WHICH SHOULD BE AVAILABLE IF ANY COMPLEX
000085                  *    MAINTENANCE OR REPAIR IS TO BE PERFORMED, IS NORMALLY MADE AS A
000086                  *    PRINTED COPY.
000087                  *
000088                  *
000089                  * ROUTINE DEMONSTRATION:
000090                  *
000091                  *    A MINIMUM SATISFACTORY TEST FOR NORMAL OPERATION MAY BE
000092                  *    OBTAINED BY SELECTING "Q" MODE OPERATION FOR EACH DEVICE PRESENT
000093                  *    AND PERMITING THE PROGRAM TO RUN AS MANY PASSES AS THERE ARE
000094                  *    DEVICES (ONE PASS PER DEVICE). THIS WILL CHECK ALL BUT THE
000095                  *    E-O-T LOGIC.
000096                  *
000097                  *
000098                  * MAIN MEMORY REQUIREMENT:
000099                  *
000100                  *    THIS PROGRAM REQUIRES 16K WORDS OF MAIN MEMORY AND WILL USE
000101                  *    ALL OF AVAILABLE MEMORY THROUGH 64K WORDS IN SAF MODE OR
000102                  *    ONE MILLION WORDS IN LAF MODE.
000103                  *
000104                  *****************************************************************
000105                  * NOTE:
000106                  *    ALL REFERENCES TO MEMORY LOCATIONS, RANGES, FILE AND RECORD COUNTS
000107                  *    ARE IN HEXADECIMAL NOTATION. REFERENCES WITHIN THE LISTING TO
000108                  *    INTERRUPT LEVELS AND ERROR AND PASS COUNTS ARE IN DECIMAL NOTATION.
000109                  *
000110                  *
000111                  * OPERATION:
000112                  *    LOAD THE PROGRAM AND START (OR RESTART) AT LOCATION
000113                  *    0100 HEX.  IF A CONSOLE IS PRESENT, VERIFY CORRECT
000114                  *    PROGRAM IDENTIFICATION FROM THE LISTING OF THE I/O
000115                  *    EQUIPMENT BY CHANNEL NUMBER AND ID-CODE. THIS LISTING WILL
000116                  *    BE OMITTED ON RESTARTS. THE LISTING WILL BE FOLLOWED BY A
000117                  *    SUMMARY OF THE ERROR REPORT DATA FORMAT.
000118                  *
000119                  *    THE CONSOLE SEARCH RULES ARE:  FIND THE CONSOLE WITH THE LOWEST  CHANNEL
000120                  *    NUMBER CONNECTED THRU AN  MDC  CONTROLLER.  IF THERE IS NO CONSOLE ON AN
000121                  *    MDC, THEN SEARCH FOR A TERMINAL WITH THE HIGHEST CHANNEL NUMBER ASSIGNED
000122                  *    TO AN ACLA ADAPTER ON AN MLC CONTROLLER.  IF NO ASYNC ADAPTER  IS FOUND,
000123                  *    THEN GO TO THE FULL CONTROL PANEL.
000124                  *
000125                  *    THERE ARE THREE CONSOLE CHANNEL OPTIONS DETERMINED BY THE VALUE  OF  LO-
000126                  *    CATION "ZV$TTY".
000127                  *
000128                  *    IF ZV$TTY EQUALS (0000), SEARCH FOR A CONSOLE.
000129                  *    IF ZV$TTY EQUALS (FFFF), ASSUME THERE IS NO CONSOLE.
000130                  *    IF ZV$TTY EQUALS NEITHER (0000), NOR (FFFF), THEN IT IS THE CONSOLE CHAN-
000131                  *    NEL NUMBER.    NOTE:  DEFAULT IS TO SEARCH FOR A CONSOLE.
000132                  *
000133                  *    ALL CONSOLE I/O IS EVEN PARITY.  IF CONSOLE IS ON MLC,  IT MUST BE ASYNC
000134                  *    AND THE BAUD RATE SET AT 1200 TO MATCH THE PROGRAM SUPPLIED RATE.  IF IT
000135                  *    IS NECESSARY TO CHANGE THE PROGRAM BAUD RATE,  THEN THE  NEW  BAUD RATE
000136                  *    CODE SHOULD BE PUT INTO LOCATION "ZV$BUD" IN HEX. THE TERMINAL BAUD RATE
000137                  *    MUST BE  SET  TO MATCH THIS NEW BAUD RATE.  THE CORRECT HEX VALUE MAY BE
000138                  *    OBTAINED FROM THE FOLLOWING TABLE.
000139                  *
000140                  *-----------------------------------------------*
000141                  *                                               *
000142                  *                BAUD  RATE  TABLE              *
000143                  *                                               *
000144                  *-----------------------------------------------*
000145                  *        ACLA I.D.  (3118, 2118, 2110)  (2108)
000146                  *        BAUD-RATE
000147                  *           50          0                   1
000148                  *           75          1                   2
000149                  *          110          2                   3
000150                  *          134          3                   4
000151                  *          150          4                   5
000152                  *          200          5                  ---
000153                  *          300          6                   6
000154                  *          600          7                   7
000155                  *          900         ---                  8
000156                  *         1050          8                  ---
000157                  *         1200          9                   9
000158                  *         1800         10 (A)              10 (A)
000159                  *         2000         11 (B)              ---
000160                  *         2400         12 (C)              11 (B)
000161                  *         3600         ---                 12 (C)
000162                  *         4800         13 (D)              13 (D)
000163                  *         7200         ---                 14 (E)
000164                  *         9600         14 (E)              15 (F)
000165                  *        19200         15 (F)              ---
000166                  *
```

```
000167                          *     TO MAKE ANY OF THE ABOVE CHANGES,  LOAD AND HALT THE PROGRAM BEFORE EX-
000168                          *     ECUTION.  INSERT CHANGE THEN EXECUTE.  MEMORY LOCATIONS OF "ZV$TTY" AND
000169                          *     "ZV$BUD" MAY BE FOUND IN MAP AT END OF LISTING.
000170                          *     CONSULT LEVEL-6 T&V MANUAL "AW94" FOR DETAILS ON HOW TO LOAD THE TESTS.
000171                          *
000172                          *     THE FOLLOWING IS A TYPICAL RESULT OF LOADING THE PROGRAM
000173                          *     AND STARTING TO RUN.
000174                          *
000175                          *               MTUX3 REV X, 9-TK PE/NRZI MAG TAPE TEST, DD MMM YY
000176                          *               ZV$LIB  REV. X.X
000177                          *               ZV$AF= X
000178                          *               WDT
000179                          *               CHAN   DEVC   ID
000180                          *               0400   DSKT   2010
000181                          *               0480   DSKT   2010
000182                          *               0500   CONS   2019
000183                          *               1400   MT-9   2046
000184                          *               1480   MT-9   2046
000185                          *               4580   CDR    2008
000186                          *               MEMORY LOW   0000XXXX
000187                          *               MEMORY HIGH  00007FFF   32K
000188                          *
000189                          *
000190                          *     THE FIRST EXECUTION OF THE PROGRAM WILL ASK THE QUESTION:
000191                          *
000192                          *               POWER FREQ, HZ ?: 60 C/R    (ON A 6/30 SYSTEM ONLY)
000193                          *
000194                          *     RESPOND WITH THE POWER LINE FREQUENCY IN HERTZ,
000195                          *     USUALLY 60 IN THE UNITED STATES AND 50 ELSEWHERE.  THE
000196                          *     PROGRAM WILL THEN CALIBRATE THE CPU CLOCK AGAINST THE
000197                          *     REAL-TIME-CLOCK. DEFAULT VALUE IS 60 HZ. WITH A 6/40,
000198                          *     THIS QUESTION WILL NOT BE ASKED.
000199                          *
000200                          *     THE PROGRAM WILL THEN ASK:
000201                          *
000202                          *               CHANNEL ?: 1400 C/R
000203                          *               DRIVE "0" IS  NRZI
000204                          *               DRIVE "1" IS  PE
000205                          *               DRIVE "3" IS  OFF LINE
000206                          *               BDCX FIRMWARE REV     XX
000207                          *
000208                          *     A DEVICE WHICH DOES NOT RESPOND WILL BE OMITTED FROM THE ABOVE
000209                          *     REPORT (IE. DRIVE "2").
000210                          *     THE RESPONSE SHOULD BE THE FOUR DIGIT HEX CHANNEL NUMBER ASSIGNED
000211                          *     TO ANY DEVICE IN THE SUBSYSTEM.  THE PROGRAM WILL USE
000212                          *     THE CONTROLLER PORTION OF THE CHANNEL NUMBER TO ADDRESS
000213                          *     ALL DEVICES. DEFAULT VALUE IS HEX 1400. "NRZI", "PE" OR "OFF LINE"
000214                          *     INDICATIONS SHOULD BE VERIFIED BY THE OPERATOR.
000215                          *
000216                          *     WHEN CHANGING ONE OR MORE DRIVES FROM NRZI TO P.E. OR VICE-
000217                          *     VERSA, MODE "I" (SEE BELOW) MUST BE RE-RUN. THE RESPONSE
000218                          *     TO "CHANNEL ?:" MAY THEN BE A "C/R".
000219                          *
000220                          *     THE ABOVE SEQUENCE WILL ONLY OCCUR ON THE FIRST START AFTER
000221                          *     A FRESH LOAD OF THE PROGRAM.  SUBSEQUENT RESTARTS WILL PICK UP
000222                          *     AT THE QUESTION "NEXT ?" WHICH IS EXPLAINED BELOW.
000223                          *
000224                          *     THE PROGRAM WILL THEN RESPOND WITH THE FOLLOWING:
000225                          *
000226                          *               M0 M1 M2 M3   NEXT ?: MX,MY,MZ...C/R
000227                          *
000228                          *     WHERE:    M0 M1 M2 M3 ARE THE MODES CURRENTLY SELECTED FOR
000229                          *               TESTING ON UNITS 0, 1, 2 AND 3 RESPECTIVELY.
000230                          *               (SEE "MODES" BELOW.)
000231                          *
000232                          *               MX, MY, MZ...ETC. ARE THE MODES WHICH THE OPERATOR WISHES
000233                          *               TO RUN ON UNITS X, Y, Z...ETC.
000234                          *
000235                          ********************************************************************
000236                          * MODES:
000237                          *     THE PROGRAM MAY OPERATE IN ANY ONE OF THE FOLLOWING MODES:
000238                          *
000239                          *          E - REPORT ERRORS AS THEY OCCUR
000240                          *          F - ENTER NEW FILE LIMIT
000241                          *          I - ID TEST, RECONFIGURE CHANNEL ADDRESSES
000242                          *          P - PRINT SUMMARY OF FIRST TEN ERRORS
000243                          *          S - CANCEL ALL ENTRIES, RESTART AT '100', ENABLE ERRORS.
000244                          *          X - SUPPRESS ERROR REPORTS
000245                          *          Z - QLF TEST MODE
000246                          *
000247                          *     THE ABOVE ARE NOT SPECIFIC TO ANY UNIT AND AS SUCH THEY DO
000248                          *     NOT APPEAR IN THE "M0 M1 M2 M3" MESSAGE PRECEDING "NEXT ?:".
000249                          *
000250                          *
000251                          *          AN - AUTOMATIC TEST MODE, UNIT "N"
000252                          *          DN - DEBUG MODE, UNIT "N"
000253                          *          QN - QUICK VERSION OF "AN"
000254                          *          RN - READ AND CHECK PRE-RECORDED TAPE, UNIT "N"
000255                          *          WN - WRITE-ONLY FULL TAPE, UNIT "N"
000256                          *          -N - DELETE UNIT "N" FROM TEST PROCEDURE LIST
000257                          *
000258                          *     THESE MODES REFER TO A PARTICULAR UNIT, 0-3, AS DESCRIBED
000259                          *     UNDER "OPERATION" ABOVE, FOLLOWING "NEXT ?:". MODE "D" IS
000260                          *     EXCLUSIVE TO UNIT "N" AND NO OTHER UNITS CAN BE
000261                          *     SELECTED SIMULTANEOUSLY. WHEN RETURNING TO "NEXT ?:" FROM
000262                          *     MODE "D", "DN" WILL NOT APPEAR IN THE REPORT OF THE SELECTED
000263                          *     MODES.
000264                          *
000265                          *
000266                          *     MODES "A" "Q" "R" AND "W" ARE COMPATIBLE WITH EACH OTHER
000267                          *     AND MAY BE RUN CONCURRENTLY. IN THIS EVENT, A SUB-TEST WILL
000268                          *     RUN ON ONE UNIT. WHEN COMPLETED, THE PROGRAM WILL DETERMINE
000269                          *     IF THE NEXT UNIT HAS BEEN SELECTED FOR TESTING. IF IT HAS,
000270                          *     THE NEXT SUB-TEST SCHEDULED FOR THAT UNIT WILL BE RUN.
000271                          *     (SEE SUB-TESTS LISTED UNDER "MODE A, USE", BELOW.)
000272                          *     ADDITIONAL UNITS MAY BE SELECTED, CHANGED OR DE-SELECTED
000273                          *     AT ANY TIME WITHOUT AFFECTING THE OVERALL SCHEDULING OF
000274                          *     THE ORIGINAL TEST. (SEE EXCEPTION IN FOLLOWING PARAGRAPH.)
000275                          *
000276                          *
000277                          *     A TEST SEQUENCE MAY BE INTERRUPTED TO ALTER OR DELETE ONE
000278                          *     OF THE SELECTED MODES OR ADD A NEW MODE BY STRIKING THE CON-
000279                          *     SOLE BREAK KEY.  INTERRUPTING AN  "A2"  FOR EXAMPLE, AND RE-
```

```
000280                          *     ENTERING AN  "A2"  WILL CAUSE THE PROGRAM TO START THE  "A2"
000281                          *     SEQUENCE FROM THE BEGINNING.  OTHERWISE, AN INTERRUPTED TEST
000282                          *     WILL RESUME AFTER C/R IS GIVEN.  GIVING A C/R ALONE WILL RESUME
000283                          *     THE PROGRAM FROM WHERE IT WAS INTERRUPTED. MODE "D".
000284                          *     IF INTERRUPTED, CANNOT BE RE-ENTERED BY MEANS OF A C/R. SHOULD
000285                          *     A BREAK OCCUR DURING DEBUG MODE, THE PROGRAM WILL RETURN TO
000286                          *     "NEXT ?:", BUT WITH ALL DRIVES DE-SELECTED.
000287                          *     IF A DRIVE GOES OFF LINE DURING TESTING, THAT DRIVE WILL
000288                          *     ALSO BE DE-SELECTED FROM FURTHER TESTING AFTER REPORTING THE
000289                          *     ERROR.
000290                          *
000291                          ************************************************************************
000292                          * ERROR REPORTING;
000293                          *     DETECTED ERRORS ARE REPORTED AS FOLLOWS:
000294                          *
000295                          *               ERR LABEL   LOC    UNIT FILE  REC   BUF   STAT1 STAT2
000296                          *
000297                          *               ERR ABCD @  LLLL    UU  FFFF  RRRR  BBBB  XXXX  YYYY
000298                          *
000299                          *     WHERE:
000300                          *        THE FIRST LINE (ERROR HEADING) IS PRINTED PRIOR TO THE FIRST
000301                          *        REPORTED ERROR ONLY.
000302                          *          AB = MAJOR LABEL, REFERS TO SUB-TEST BEING
000303                          *               PERFORMED (SEE "MODE A, USE")
000304                          *          CD = MINOR LABEL, INDICATES SPECIFICALLY WHERE
000305                          *               ERROR WAS DETECTED
000306                          *        LLLL = HEX LOCATION WHERE THE ERROR WAS DETECTED.
000307                          *          UU = NUMBER OF THE UNIT UNDER TEST (0-3)
000308                          *        FFFF = HEX FILE NUMBER
000309                          *        RRRR = HEX RECORD NUMBER
000310                          *        BBBB = HEX ADDRESS OF THE DATA BUFFER INVOLVED
000311                          *        XXXX = HEX STATUS WORD #1
000312                          *        YYYY = HEX STATUS WORD #2
000313                          *
000314                          *
000315                          * NOTE: THIS PROGRAM CAN BE OPERATED WITHOUT A CONSOLE. ERROR-DATA IS
000316                          *     LIMITED TO "ABCD" IN REGISTERS "R1" AND "R2", AND "LLLL" IN REGISTER
000317                          *     "R2". ALL ENTRIES MUST BE MADE VIA REGISTER "R1". THE PROCESS IS FURTHER
000318                          *     EXPLAINED IN MANUAL "Aw94" ENTITLED "LEVEL-6 SYSTEM CHECKOUT AND T&V
000319                          *     MANUAL".
000320                          *
000321                          *     IF THE ERROR INVOLVES ERRONEOUS DATA THE REPORT WILL
000322                          *     BE FOLLOWED BY FROM ONE TO EIGHT LINES OF THE FOLLOWING:
000323                          *
000324                          *        BYTE XXXX  SB  YY,  IS  ZZ    RETRY  ABORT
000325                          *
000326                          *     WHERE:
000327                          *        XXXX = RELATIVE BYTE LOCATION IN DATA BUFFER (0-07FF)
000328                          *          YY = DATA IN WRITE BUFFER (SHOULD-BE)
000329                          *          ZZ = ACTUAL DATA IN READ BUFFER AT THAT LOCATION (IS)
000330                          *       RETRY = TYPED ONLY AFTER THE LAST DATA ERROR DETECTED
000331                          *               (NOT TO EXCEED EIGHT ERRORS)
000332                          *       ABORT = TYPED ONLY AFTER THREE READ ATTEMPTS
000333                          *
```

```
000334                          /
000335                          *
000336                          *     THE FOLLOWING IS A SUMMARY OF ALL MAJOR AND MINOR LABELS AND
000337                          *     THEIR ERROR INDICATION:
000338                          *
000339                          *
000340                          *       LABEL   INDICATION                          PROBABLE CRU
000341                          *       -----   ----------                          ------------
000342                          *         GLT0  GLT NEVER RAN                       CHECK HEX ROTARY SWITCH
000343                          *         GLT2  GLT FAILURE                         BUS OR MDC
000344                          *
000345                          *         GLP0  GLT FAILURE                         ADAPTER PORT 0
000346                          *         GLP1  GLT FAILURE                         ADAPTER PORT 1
000347                          *         GLP2  GLT FAILURE                         ADAPTER PORT 2
000348                          *         GLP3  GLT FAILURE                         ADAPTER PORT 3
000349                          *
000350                          *         GL7F  MDC, INTERNAL                       MDC
000351                          *         GL9F  CAI, INDETERMINATE                  MDC OR ADAPTER
000352                          *         GLBF  CAI, MDC IMPLICATED                 MDC
000353                          *         GLDF  CAI, ADAPTER IMPLICATED             ADAPTER
000354                          *         GLEF  ADAPTER, INTERNAL                   ADAPTER
000355                          *
000356                          *         GLF7  ADAPTER, PE FUNCTION                ADAPTER
000357                          *         GLF9  DAI, INDETERMINATE                  ADAPTER OR DRU
000358                          *         GLFB  DAI, ADAPTER IMPLICATED             ADAPTER
000359                          *         GLFD  DAI, DRU IMPLICATED                 DRU
000360                          *         GLFE  DRU OR READ CLOCK                   DRU OR CLOCK
000361                          *
000362                          *
000363                          *     MAJOR LABEL (LABEL1)
000364                          *
000365                          *         AA--  WRITE FILE AND RECORD ADDRESS IN FIRST RECORD
000366                          *         AB--  WRITE, READ AND CHECK RECORD OF ZEROS
000367                          *         AC--  WRITE, READ AND CHECK 1'S IN BIT 8 OF EACH BYTE (LSB)
000368                          *         AD--  WRITE, READ AND CHECK 1'S IN BIT 7 OF EACH BYTE
000369                          *         AE--  WRITE, READ AND CHECK 1'S IN BIT 6 OF EACH BYTE
000370                          *         AF--  WRITE, READ AND CHECK 1'S IN BIT 5 OF EACH BYTE
000371                          *         AG--  WRITE, READ AND CHECK 1'S IN BIT 4 OF EACH BYTE
000372                          *         AH--  WRITE, READ AND CHECK 1'S IN BIT 3 OF EACH BYTE
000373                          *         AI--  WRITE, READ AND CHECK 1'S IN BIT 2 OF EACH BYTE
000374                          *         AJ--  WRITE, READ AND CHECK 1'S IN BIT 1 OF EACH BYTE (MSB)
000375                          *         AK--  WRITE, READ AND CHECK ALL 1'S
000376                          *         AL--  WRITE, READ AND CHECK CHECKER-BOARD PATTERN
000377                          *         AM--  WRITE, READ AND CHECK CHECKER-BOARD PATTERN, OPPOSITE PHASE
000378                          *         AN--  WRITE, READ AND CHECK RANDOM LENGTH RECORD, RANDOM DATA
000379                          *         AO--  WRITE, READ AND CHECK RANDOM LENGTH RECORD, RANDOM DATA
000380                          *         AP--  WRITE RANDOM DATA FROM STANDARD WRITE BUFFER
000381                          *         AQ--  READ PREVIOUS RECORD INTO RANDOMLY LOCATED BUFFER
000382                          *         AR--  WRITE RANDOM DATA FROM RANDOMLY LOCATED BUFFER
000383                          *         AS--  READ PREVIOUS RECORD INTO STANDARD READ BUFFER
000384                          *         AT--  WRITE (SPACE OVER) FILE MARK AT END OF DATA FILE
000385                          *         AU--  ARBITRARILY SKIP AROUND WITHIN PRECEDING FILE, CHECK DATA.
000386                          *         BF--  BACK-SPACE FILE TEST
000387                          *         BR--  BACK-SPACE RECORD TEST
000388                          *         D1--  DEBUG MODE, INITIAL SET-UP
000389                          *         DD--  DEBUG MODE, DATA COMPARISON ROUTINE
000390                          *         DE--  DEBUG MODE, ERASE
000391                          *         DM--  DEBUG MODE, WRITE A FILE MARK
000392                          *         DO--  DEBUG MODE, BACKSPACE A FILE
000393                          *         DR--  DEBUG MODE, READ A RECORD
000394                          *         DT--  DEBUG MODE, BACKSPACE A RECORD
000395                          *         DU--  DEBUG MODE, REWIND TO B-O-T
000396                          *         DV--  DEBUG MODE, FORWARD SPACE A RECORD
000397                          *         DW--  DEBUG MODE, WRITE A RECORD
000398                          *         DY--  DEBUG MODE, FORWARD SPACE A FILE
000399                          *         EF--  WRITE FILE OVER PREVIOUS FILE
000400                          *         EP--  WRITE AND CHECK END-OF-PASS RECORD.
000401                          *         ER--  ERASE TEST
000402                          *         FF--  FORWARD-SPACE FILE TEST
000403                          *         FI--  WRITE FILE MARK DELIMITER (MODE 'A')
000404                          *         FM--  FILE MARK TEST
000405                          *         FR--  FORWARD-SPACE RECORD TEST
000406                          *         MI--  CHANNEL CONFIGURATION ROUTINE
000407                          *         GL--  GLT TEST
000408                          *         RP--  INTERRUPT LOGIC AND POLLING TEST
000409                          *         RW--  REWIND TEST
000410                          *         SA--  SET AND RESET ALL STATUS BITS
000411                          *         SB--  OPERATION CHECK AND FUNCTIONALITY N/AVAILABLE (READ REVERSE)
000412                          *         SC--  READ RECORD WITH RANGE =0
000413                          *         SD--  UNEQUAL LENGTH CHECK
000414                          *         SE--  OPERATION CHECK WITH DIFFERENT DIRECTION FOR IOLD AND TASK
000415                          *         SF--  CHECK ANSI INHIBIT MODE
000416                          *         SG--  LRC, CRC, VRC CHECK
000417                          *         SH--  DATA SERVICE RATE ERROR CHECK (ONLY WITH "BDC2" FIRMWARE)
000418                          *         SI--  NONEXISTANT RESOURCE ERROR
000419                          *         T1-T9 DATA TURN-AROUND TEST (DO ONLY WITH "BDC2" FIRMWARE)
000420                          *         TI--  CALIBRATE CPU AGAINST RTC FOR TIMING TESTS
000421                          *         X0-X3 SEQUENCER TO HANDLE THE TESTING OF UP TO FOUR UNITS.
000422                          *
000423                          *
000424                          *     MINOR LABEL (LABEL2)
000425                          *
000426                          *         --01  SYSTEM SHOULDN'T BE BUSY
000427                          *         --02  SHOULD HAVE BEEN READY
000428                          *         --03  SHOULD HAVE BEEN READY FOR REWIND
000429                          *         --04  UNIT N/RDY WITH ATTN SET
000430                          *         --05  STATUS INDICATES P.E. DRIVE, ID DOESN'T
000431                          *         --06  STATUS INDICATES NRZI DRIVE, ID DOESN'T
000432                          *         --07  STATUS WORD 1 AND/OR 2 NG
000433                          *         --08  STATUS WORD 1 NG AFTER REWIND
000434                          *         --09  TURN-AROUND TEST, RANGE ERROR
000435                          *         --10  SHOULD HAVE BEEN READY FOR COMMAND
000436                          *         --11  STATUS NG DURING REWIND
000437                          *         --12  STATUS NG AFTER REWIND
000438                          *         --13  CAN'T GET STATUS, OR STATUS NG AFTER REWIND FROM BOT
000439                          *         --14  STATUS NG AFTER REWIND
000440                          *         --15  STATUS NG AFTER A WRITE OPERATION
000441                          *         --16  SHOULD HAVE BEEN READY PRIOR TO A READ OPERATION
000442                          *         --17  SHOULD HAVE BEEN READY FOR REWIND
000443                          *         --18  SYSTEM SHOULD HAVE BEEN READY WHEN ABOUT TO WRITE
000444                          *         --19  STATUS NG AFTER ERASE
000445                          *         --20  TIMING ERROR ON A 45 IPS DRIVE
000446                          *         --21  STATUS NG AFTER WRITING A FILE MARK
```

```
000447                          *        --22   TIMING ERROR ON A 75 IPS DRIVE
000448                          *        --23   RFU
000449                          *        --24   INITIALIZE DIDN'T DO IT'S THING
000450                          *        --25   STATUS NG AFTER "BACK SPACE A FILE"
000451                          *        --26   FM SHOULDN'T BE DETECTED ON NORMAL RECORD
000452                          *        --27   STATUS SHOULD BE 8010-8010 AFTER READ REVERSE
000453                          *        --28   STATUS SHOULD BE 8210-8000 FOR WRITE W/RANGE =0
000454                          *        --29   STATUS NG AFTER FORWARD SPACING A RECORD WHICH IS A FILE MARK
000455                          *        --30   STATUS NG AFTER BACK SPACING A FILE
000456                          *        --31   STATUS NG AFTER FORWARD SPACING A FILE
000457                          *        --32   STATUS WORD 1 SB FM, N/BOT
000458                          *        --33   SHOULDN'T TAKE ANY TIME TO BACK SPACE RECORD AT BOT
000459                          *        --34   STATUS WORD 1 NG AFTER BACKSPACE RECORD FROM BOT
000460                          *        --35   SHOULDN'T TAKE ANY TIME TO BACK SPACE A FILE AT BOT
000461                          *        --36   SHOULD HAVE INTERRUPTED BUT DIDN'T
000462                          *        --37   INTERRUPTED WHEN IT SHOULDN'T HAVE
000463                          *        --38   DIDN'T INTERRUPT WHEN CP LEVEL WENT TO 63
000464                          *        --39   INTERRUPTING DEVICE WAS NOT A TAPE
000465                          *        --40   "S" REGISTER IN SAVE AREA IS INCORRECT
000466                          *        --41   ACTIVITY FLAG IS ON WHEN IT SHOULDN'T BE
000467                          *        --42   ACTIVITY FLAG IS OFF WHEN IT SHOULD BE ON
000468                          *        --43   CPU TRIED TO INTERRUPT
000469                          *        --44   STATUS NG AFTER BACK SPACE A FILE FROM BOT
000470                          *        --45   STATUS SHOULD BE FCFF-8FFF
000471                          *        --46   STATUS SHOULD BE BCF9-8FFF
000472                          *        --47   STATUS SHOULD BE FFFF-8FFF
000473                          *        --48   STATUS SHOULD BE 8080-8000 FOR READ W/RANGE =0
000474                          *        --49   RANGE SHOULD BE =0, DATA SHOULD BE 3333
000475                          *        --50   STATUS SHOULD BE 8080-8000, RANGE SHOULD BE =1, LAST DATA SB =33
000476                          *        --51   STATUS SHOULD BE 8080-8000, RANGE SB =0, LAST BYTE+1 SB =33
000477                          *        --52   STATUS SHOULD BE 8210-8000 FOR WRITE/READ DIRECTION CONFLICT
000478                          *        --53   STATUS SHOULD BE 8000-8000 AFTER ANSI INHIBIT WRITE
000479                          *        --54   STATUS SHOULD BE 8C00-8000 FOR NON-ANSI READ OF SHORT RECORD
000480                          *        --55   STAT SHOULD BE 8000-8000 FOR READ SHORT RECORD W/ANSI INHIBITED
000481                          *        --56   STAT SHOULD BE A040-8602 AFTER DIAG WRITE W/FAKE CRC & LRC GAPS
000482                          *        --57   STATUS SHOULD BE A000-8700 AFTER READING DIAGNOSTIC RECORD
000483                          *        --58   DATA SERVICE RATE ERROR NOT DETECTED
000484                          *        --59   STATUS SHOULD INDICATE "FUNCTIONALITY NOT AVAILABLE"
000485                          *        --60   IN MIDDLE OF OPERATION, SHOULDN'T GET ACK'ED
000486                          *        --61   DIDN'T GET INTERRUPT WHEN EXPECTED
000487                          *        --62   DEVICE SHOULDN'T HAVE INTERRUPTED WHEN IT'S RUPT LEVEL =0
000488                          *        --63   OUTPUT CONTROL TO STOP IO SHOULD ALWAYS BE ACK'ED
000489                          *        --64   OPERATION TIMED OUT, SHOULD BE LESS THAN 5 SECONDS
000490                          *        --65   STATUS SHOULD BE XXXX-8000 AFTER TASK OUTPUT
000491                          *               (XXXX=8000 FOR BDC2, =C000 FOR BDC3 FIRMWARE)
000492                          *        --66   MISSING RESOURCE TRAP, RESTART AT "NEXT ?:".
000493                          *        --67   INVALID SPEED INDICATED BY ID CODE
000494                          *        --DA   DATA COMPARISON ERROR
000495                          *        --DO   DMA OVERFLOW
000496                          *        --E1   STATUS NG AFTER ERASE OPERATION
000497                          *        --F1   STATUS NG AFTER FORWARD SPACE A FILE AND WAIT
000498                          *        --F4   STATUS NG AFTER BACK SPACE A FILE AND WAIT
000499                          *        --ND   NO DEVICE ON LINE
000500                          *        --R1   STATUS NG AFTER FORWARD SPACE A RECORD AND WAIT
000501                          *        --R4   STATUS NG AFTER BACK SPACE A RECORD AND WAIT
000502                          *
000503                          *
000504                          *    THIS PROGRAM CAN BE OPERATED WITHOUT A CONSOLE PRESENT.
000505                          *    UNDER THESE CIRCUMSTANCES ERROR DATA IS LIMITED TO THE MAJOR
000506                          *    AND MINOR ERROR LABELS DISPLAYED IN HARDWARE REGISTERS
000507                          *    "R1" AND "R2" RESPECTIVELY, AND "LLLL" DISPLAYED IN
000508                          *    REGISTER "B2". ALL PARAMETER ENTRIES TO THE PROGRAM MUST BE MADE
000509                          *    VIA REGISTER "R1". THE PROCESS IS FURTHER EXPLAINED IN
000510                          *    MANUAL "AW-94" ENTITLED "LEVEL-6 SYSTEM CHECKOUT AND
000511                          *    T&V MANUAL".
000512                          *
000513                          *    BEFORE ATTEMPTING TO READ, THE READ BUFFER IS FILLED
000514                          *    WITH 33 (00110011) IN ALL BYTES.  A FAILURE TO TRANSFER
000515                          *    DATA IS INDICATED BY YY BEING 33 HEX.
000516                          *    IF DATA ERRORS ARE DETECTED DURING THE NORMAL DATA TRANSFER
000517                          *    TESTS, THE PROGRAM WILL ATTEMPT TO RE-READ THE
000518                          *    RECORD UP TO TWO MORE TIMES.  IF THE RETRY IS SUCCESSFUL,
000519                          *    THE PROGRAM WILL GO ON WITH THE NEXT TEST.  IF NOT SUCCESS-
000520                          *    FUL, IT WILL TYPE "ABORT" AND THEN GO ON WITH THE NEXT TEST
000521                          *    ANYWAY.
000522                          *
000523                          *    IF AT ANY TIME THE OPERATOR WISHES TO SUPPRESS ERROR
000524                          *    REPORTING, HE MAY DO SO BY SELECTING MODE "A". REPORTING
000525                          *    OF ERRORS MAY BE RESUMED BY GOING TO "NEXT" AND SELECTING
000526                          *    MODE "E". AT LOAD TIME, THE DEFAULT MODE IS TO REPORT ERRORS.
000527                          *
000528                          ************************************************************************
000529                          * END-OF-PASS REPORT:
000530                          *    AFTER EITHER "N" DATA FILES HAVE BEEN WRITTEN (SEE "MODE F, USE")
000531                          *    OR "E-O-T" HAS BEEN
000532                          *    DETECTED ON ANY ONE OF THE SELECTED UNITS AN "END-OF-PASS"
000533                          *    RECORD IS WRITTEN ON TAPE AND THE FOLLOWING MESSAGE IS TYPED:
000534                          *
000535                          *       MODE MN  PASS PPP   EEEE ERR(S)   FFFF HEX FILE(S)...
000536                          *
000537                          *    WHERE:
000538                          *        MN = CURRENT MODE AND UNIT (EG. "A0", "W1", "R2" ETC.)
000539                          *       PPP = NUMBER OF PASSES FOR THAT UNIT SINCE START OF MODE (DEC)
000540                          *      EEEE = TOTAL DEC NUMBER OF ERRORS ACCUMULATED (ALL UNITS) SINCE
000541                          *             START OF TEST (START OF TEST IS DEFINED AS THE TIME
000542                          *             WHEN A MODE IS SELECTED ON ONE OR MORE UNITS AND
000543                          *             NO OTHER UNITS HAD BEEN PREVIOUSLY SELECTED, IE.
000544                          *             FOLLOWING "STOP")
000545                          *       FFFF = TOTAL NUMBER OF DATA FILES WRITTEN ON THIS TAPE FOR
000546                          *              THIS PASS (HEX)
000547                          *
000548                          *    AVERAGE NUMBER OF DATA BYTES TRANSFERRED PER FILE.
000549                          *
000550                          *                          MODES A/Q  MODES A/Q  MODE W     MODE R
000551                          *                          WRITTEN    READ       WRITTEN    READ
000552                          *
000553                          *     MODE A, PER FILE      14180      20823      14180      14180
000554                          *     MODE Q, PER PASS     113440     166824       ---        ---
000555                          *
000556                          *
000557                          ************************************************************************
```

```
000556                  /*****************************************************************
000559                  * STATUS WORDS:
000560                  *     THE TWO STATUS WORDS HAVE THE FOLLOWING SIGNIFICANCE:
000561                  *
000562                  *     FIRST STATUS WORD
000563                  *       BIT  STATUS
000564                  *       ------------------------
000565                  *        0     READY
000566                  *        1     ATTENTION
000567                  *        2     RETRYABLE MEDIA ERROR
000568                  *        3     RFU - MBZ
000569                  *       ------------------------
000570                  *        4     CORRECTED MEDIA ERROR
000571                  *        5     TAPE MARK DETECTED
000572                  *        6     E-O-T
000573                  *        7     E-O-T
000574                  *       ------------------------
000575                  *        8     UNEQUAL LENGTH CHECK
000576                  *        9     NON-RETRYABLE ERROR
000577                  *       10     RFU - MBZ
000578                  *       11     OPERATION CHECK
000579                  *       ------------------------
000580                  *       12     CORRECTED MEMORY ERROR
000581                  *       13     NON-EXISTANT RESOURCE ERROR
000582                  *       14     BUS PARITY ERROR
000583                  *       15     UNCORRECTED MEMORY ERROR
000584                  *
000585                  *
000586                  *     SECOND STATUS WORD
000587                  *       BIT  STATUS
000588                  *       ------------------------
000589                  *        0     ON LINE
000590                  *        1     REWINDING
000591                  *        2     FILE IN PROTECT
000592                  *        3     HIGH DENSITY SELECTED
000593                  *       ------------------------
000594                  *        4     DATA SERVICE RATE ERROR
000595                  *        5     UNCORRECTABLE CHARACTER ERROR
000596                  *        6     NRZI CRC ERROR / P.E. SINGLE TRACK ERROR
000597                  *        7     NRZI LRC ERROR / P.E. MULTIPLE TRACK ERROR
000598                  *       ------------------------
000599                  *        8     ID BURST AREA ERROR
000600                  *        9     RFU - MBZ
000601                  *       10     TIME-OUT CHECK
000602                  *       11     FUNCTIONALITY NOT AVAILABLE
000603                  *       ------------------------
000604                  *       12     BEGINNING-OF-BLOCK-EARLY ERROR
000605                  *       13     BEGINNING-OF-BLOCK-LATE ERROR
000606                  *       14     END-OF-BLOCK-EARLY ERROR
000607                  *       15     END-OF-BLOCK-LATE ERROR
000608                  *
000609                  *****************************************************************
000610                  * MODE "A", USE
000611                  *     AS SOON AS A C/R IS RECEIVED BY THE COMPUTER THE PROGRAM WILL
000612                  *     START ON THE FIRST OF THE "SELECTED" DRIVES AND WILL PERFORM
000613                  *     THE FIRST OF THE SUB-TESTS LISTED BELOW ON THAT DRIVE. IT WILL
000614                  *     THEN GO ON TO THE NEXT SELECTED DRIVE AND PERFORM THE FIRST
000615                  *     SUB-TEST ON IT, AND SO ON. IF A SELECTED DRIVE IS REWINDING THAT
000616                  *     DRIVE WILL BE SKIPPED AND THE PROGRAM WILL CONTINUE TESTING THE
000617                  *     REMAINING DRIVES UNTIL THE REWIND IS COMPLETE.
000618                  *
000619                  *     DURING THE OPERATION OF MODE "A" THE FOLLOWING SUB-TESTS ARE
000620                  *     PERFORMED. THE TWO LETTER "MAJOR LABEL" WILL BE INCLUDED IN THE
000621                  *     MESSAGE WHENEVER AN ERROR IS REPORTED DURING THAT TEST. SPECIFIC
000622                  *     TEST SECTIONS MAY BE FOUND IN THE LISTING BY LOCATING THE
000623                  *     REPORTING LABEL IN THE ACCOMPANYING CROSS-REFERENCE LIST.
000624                  *     THE FIRST GROUP OF TESTS IS PERFORMED ONLY ONCE AT THE START
000625                  *     OF THE FIRST PASS THROUGH THE TAPE. THESE ARE SPECIAL PURPOSE
000626                  *     TESTS DESIGNED TO INTRODUCE ERRORS TO DETERMINE IF THE SYSTEM
000627                  *     IS FUNCTIONING PROPERLY.
000628                  *
000629                  *     MAJOR LABEL
000630                  *        X0-X3   SEQUENCER TO HANDLE THE TESTING OF UP TO FOUR UNITS.
000631                  *                THIS HANDLER RUNS CONTINUOUSLY THROUGHOUT THE TEST
000632                  *        T1-T9   DATA TURN-AROUND TEST (DO ONLY WITH "BDC2" FIRMWARE)
000633                  *        RW   -  REWIND TEST
000634                  *        ER   -  ERASE TEST
000635                  *        FM   -  FILE MARK TEST
000636                  *        BR   -  BACK-SPACE RECORD TEST
000637                  *        FR   -  FORWARD-SPACE RECORD TEST
000638                  *        BF   -  BACK-SPACE FILE TEST
000639                  *        FF   -  FORWARD-SPACE FILE TEST
000640                  *        EF   -  WRITE FILE OVER PREVIOUS FILE
000641                  *        RP   -  INTERRUPT LOGIC AND POLLING TEST
000642                  *        SA-SI   STATUS WORD TESTS
000643                  *
000644                  *     FOLLOWING THESE PRELIMINARY TESTS, THE PROGRAM WILL BEGIN THE
000645                  *     GENERAL WRITE/READ DATA ROUTINES AS OUTLINED BELOW:
000646                  *
000647                  *        FI   -  WRITE FILE MARK
000648                  *        AA   -  WRITE FILE AND RECORD ADDRESS IN FIRST RECORD
000649                  *        AB   -  WRITE/READ RECORD OF ZEROS
000650                  *        AC   -  W/R 1'S IN BIT 8 OF EACH BYTE (LSB)
000651                  *        AD   -  W/R 1'S IN BIT 7 OF EACH BYTE
000652                  *        AE   -  W/R 1'S IN BIT 6 OF EACH BYTE
000653                  *        AF   -  W/R 1'S IN BIT 5 OF EACH BYTE
000654                  *        AG   -  W/R 1'S IN BIT 4 OF EACH BYTE
000655                  *        AH   -  W/R 1'S IN BIT 3 OF EACH BYTE
000656                  *        AI   -  W/R 1'S IN BIT 2 OF EACH BYTE
000657                  *        AJ   -  W/R 1'S IN BIT 1 OF EACH BYTE (MSB)
000658                  *        AK   -  W/R ALL 1'S
000659                  *        AL   -  W/R CHECKER-BOARD PATTERN
000660                  *        AM   -  W/R CHECKER-BOARD PATTERN, OPPOSITE PHASE
000661                  *        AN   -  W/R RANDOM LENGTH RECORD, RANDOM DATA
000662                  *        AO   -  W/R RANDOM LENGTH RECORD, RANDOM DATA
000663                  *        AP   -  WRITE RANDOM DATA FROM STANDARD WRITE BUFFER
000664                  *        AQ   -  READ PREVIOUS RECORD INTO RANDOMLY LOCATED BUFFER
000665                  *                UP TO 64K. (IF ONLY 8K, USE STANDARD READ BUFFER)
000666                  *        AR   -  WRITE RANDOM DATA FROM RANDOMLY LOCATED BUFFER
000667                  *                UP TO 64K. (IF ONLY 8K, USE STANDARD WRITE BUFFER)
000668                  *        AS   -  READ PREVIOUS RECORD INTO STANDARD READ BUFFER
000669                  *        AT   -  WRITE (SPACE OVER) A FILE MARK
000670                  *
```

```
000671                  *     AFTER THE FILE MARK IS WRITTEN, THE FOLLOWING TEST IS PERFORMED.
000672                  *
000673                  *        AU  -  ARBITRARILY SKIP AROUND WITHIN PRECEDING FILE,
000674                  *               CHECK DATA.
000675                  *
000676                  *     IF AN "END-OF-PASS" IS DETERMINED TO EXIST (IE. END-OF-TAPE OR "N"
000677                  *     DATA FILES DONE (SEE BELOW)) AN "END-OF-PASS" RECORD IS WRITTEN ON
000678                  *     TAPE AND THE "END-OF-PASS" MESSAGE WILL BE TYPED OUT ON THE CONSOLE.
000679                  *
000680                  *        EP  -  WRITE AND CHECK END-OF-PASS RECORD.
000681                  *
000682                  *     IF NOT END-OF-PASS, TESTING IS RESUMED WITH TEST "AA"
000683                  *
000684                  *     AFTER TYPING THE END-OF-PASS MESSAGE, THE PROGRAM WILL
000685                  *     RESUME TESTING WITH TEST "AA" FROM B-O-T.
000686                  *     DURING ALL OF THE ABOVE, ROUTINES X0-X3 ARE EXECUTING
000687                  *     CONSECUTIVELY, SEQUENCING BETWEEN ALL OF THE SELECTED UNITS.
000688                  *     (NOTE: THE SEQUENCE "AA-AU" IS CONSIDERED TO BE A SINGLE
000689                  *     TEST WHICH MUST BE COMPLETED BEFORE A TEST MAY BE PERFORMED
000690                  *     ON THE NEXT DRIVE.)
000691                  *
000692                  *  DATA STRUCTURE:
000693                  *     AFTER THE PRELIMINARY TESTS ARE COMPLETED OR SKIPPED, EACH
000694                  *     FILE WILL CONTAIN 17 (HEX 11) RECORDS, (TESTS AA-AS AND AU).
000695                  *     THE FIRST WORD (BYTES 0,1) WILL CONTAIN THE FILE NUMBER AND
000696                  *     THE SECOND WORD (BYTES 2,3) WILL CONTAIN THE RECORD NUMBER.
000697                  *     THE REMAINING WORDS MAY BE A FIXED OR RANDOM DATA PATTERN
000698                  *     OF FIXED OR RANDOM LENGTH. THE TABLE BELOW SHOWS THE FIRST
000699                  *     SIX BYTES OF EACH OF THE RECORDS IN FILE 11 (DEC) AFTER HAVING
000700                  *     WRITTEN AT LEAST THAT MANY FILES.
000701                  *
000702                  *        HEX                                          DEC
000703                  *        REC    0    1    2    3    4    5   DATA    BYTES  LABEL
000704                  *        ---   --   --   --   --   --   --   ------  -----  -----
000705                  *          0   00   0B   00   00   00   00   FIXED      18   AA
000706                  *          1   00   0B   00   01   00   00   FIXED     275   AB
000707                  *          2   00   0B   00   02   01   01   FIXED    1900   AC
000708                  *          3   00   0B   00   03   02   02   FIXED    1727   AD
000709                  *          4   00   0B   00   04   04   04   FIXED    1004   AE
000710                  *          5   00   0B   00   05   08   08   FIXED     131   AF
000711                  *          6   00   0B   00   06   10   10   FIXED     804   AG
000712                  *          7   00   0B   00   07   20   20   FIXED    1435   AH
000713                  *          8   00   0B   00   08   40   40   FIXED     312   AI
000714                  *          9   00   0B   00   09   80   80   FIXED      29   AJ
000715                  *          A   00   0B   00   0A   FF   FF   FIXED    1967   AK
000716                  *          B   00   0B   00   0B   A9   56   CHECKER   408   AL
000717                  *          C   00   0B   00   0C   56   A9   CHECKER    92   AM
000718                  *          D   00   0B   00   0D   20   55   RANDOM  RANDOM   AN
000719                  *          E   00   0B   00   0E   21   D5   RANDOM  RANDOM   AO
000720                  *          F   00   0B   00   0F   18   DA   RANDOM    1024   AP, AQ
000721                  *         10   00   0B   00   10   16   5A   RANDOM    1024   AR, AS
000722                  *
000723                  ************************************************************************
000724                  *  MODE "D", USE
000725                  *     THE DEBUG MODE ALLOWS THE OPERATOR TO LINK TOGETHER
000726                  *     AND EXECUTE A NUMBER OF DIFFERENT SUBROUTINES FOR THE
000727                  *     PURPOSE OF LOCATING SPECIFIC HARDWARE PROBLEMS.
000728                  *     THE AVAILABLE SUBROUTINES ARE:
000729                  *
000730                  *     LINK
000731                  *        A - PRINT CONTENTS OF WRITE BUFFER
000732                  *        B - PRINT CONTENTS OF READ BUFFER
000733                  *        C - INITIALIZE
000734                  *        D - COMPARE READ BUFFER TO WRITE BUFFER
000735                  *        E - ERASE A BLOCK OF TAPE
000736                  *        F - FILL WRITE BUFFER, FIXED DATA
000737                  *        G - FILL WRITE BUFFER, RANDOM DATA
000738                  *        H - HALT EXECUTION, RETURN TO DEBUG
000739                  *        I - FILL WRITE BUFFER, ASCENDING COUNT
000740                  *        J - ESTABLISH RANDOM WRITE BUFFER UP TO 64K
000741                  *        K - ESTABLISH RANDOM READ BUFFER UP TO 64K
000742                  *        L - SPARE SUBROUTINES (NOP'S)
000743                  *        M - WRITE A TAPE MARK
000744                  *        N - GENERATE RANDOM PARAMETERS (LENGTH, DATA)
000745                  *        O - BACK-SPACE A FILE
000746                  *        P - PRINT PARAMETER SET (UNIT, HEX LENGTH, DATA, W/BUF, R/BUF)
000747                  *        Q - TRANSFER READ BUFFER TO WRITE BUFFR
000748                  *        R - READ A RECORD (DON'T CHECK DATA)
000749                  *        S - INPUT BOTH STATUS WORDS
000750                  *        T - BACK-SPACE A RECORD
000751                  *        U - REWIND TO "B-O-T"
000752                  *        V - FORWARD-SPACE A RECORD
000753                  *        W - WRITE A RECORD
000754                  *        X - EXIT TO "NEXT ?:"
000755                  *        Y - FORWARD-SPACE A FILE
000756                  *        Z - PRINT STATUS, RANGE, FILE NUMBER, RECORD NUMBER. (ALL HEX)
000757                  *
000758                  *     TO MANUALLY SELECT A PARTICULAR WRITE OR READ BUFFER
000759                  *     LOCATION, ENTER THE BUFFER ADDRESS IN LOCATION "JJPT"
000760                  *     OR "KKPT" RESPECTIVELY AFTER ENTERING THE DEBUG MODE.
000761                  *
000762                  *     AFTER TYPING "DN", WHERE N IS THE DRIVE NUMBER, THE COMPUTER WILL
000763                  *     REQUEST A SET OF PARAMETERS BY TYPING THE FOLLOWING:
000764                  *
000765                  *        RANGE, DATA ?: LLL,DD C/R
000766                  *
000767                  *     TO ENTER NEW PARAMETERS THE RESPONSE SHOULD BE AS FOLLOWS:
000768                  *
000769                  *         LLL = RECORD LENGTH (12-800 HEX BYTES, DEFAULT = 800 AT LOAD TIME,
000770                  *               OR WHATEVER THE MOST RECENT VALUE WAS AS A RESULT OF
000771                  *               RUNNING MODES 'A', 'Q', 'W' OR 'R')
000772                  *
000773                  *               SPECIAL PROVISIONS HAVE BEEN MADE IN ORDER TO
000774                  *               READ TAPES GENERATED EXTERNALLY. ENTERING A
000775                  *               RANGE OF ZERO WILL CAUSE THE PROGRAM TO DEFAULT
000776                  *               TO A RANGE OF '800 AND WILL SUPPRESS THOSE
000777                  *               ERRORS SPECIFICALLY CAUSED BY A READ OPERATION
000778                  *               WHICH RESULTED IN AN "UNEQUAL LENGTH CHECK".
000779                  *               TAPES MAY THEN BE READ (WITHOUT REQUIRING A
000780                  *               WRITE ENABLE RING) TO DETERMINE IF OTHER
000781                  *               ERRORS ARE PRESENT.
000782                  *
000783                  *               (NOTE! RANGES FROM HEX 1 TO 11 MAY BE USED IN READ MODE.
```

```
000784                          *               HOWEVER, SINCE THIS DEFIES THE ANSI STANDARD, ERRORS
000785                          *               WILL BE REPORTED.)
000786                          *
000787                          *         DD = DATA PATTERN  (0-FF HEX, DEFAULT = FF)
000788                          *
000789                          *     THE TWO PARAMETERS ARE SEPARATED BY A COMMA. IF NO VALUES ARE
000790                          *     ENTERED BEFORE EITHER DELIMITER THAT PARAMETER WILL REMAIN
000791                          *     UNCHANGED FROM THE LAST TIME IT WAS ENTERED. A RESPONSE OF ONLY
000792                          *     A CARRIAGE RETURN WILL LEAVE BOTH PARAMETERS UNCHANGED. THE DATA
000793                          *     PATTERN BYTE WILL BE REPEATED THROUGHOUT THE ENTIRE RECORD
000794                          *     LENGTH ONLY WHEN LINK 'F' (SEE BELOW) IS EXECUTED.
000795                          *     A "BREAK" WILL CAUSE A RETURN TO "NEXT ?:".
000796                          *     IN ANY CASE, A CARRIAGE RETURN WILL CAUSE THE FOLLOWING:
000797                          *
000798                          *        LINKS ?: XXXX...C/R
000799                          *
000800                          *     THE RESPONSE TO THIS REQUEST FOR SUBROUTINE LINKS SHOULD CONSIST
000801                          *     OF A COMBINATION OF UP TO TWENTY (20) OF THE LINKS DESCRIBED
000802                          *     BELOW AND INDICATED BY "X'S" ABOVE. THE STRING MUST BE
000803                          *     TERMINATED BY A C/R. TO RETURN FROM "LINKS" TO "RANGE, DATA", TYPE
000804                          *     "H" C/R. TO ABORT AND RETURN TO "NEXT", TYPE "BREAK". AT THIS POINT,
000805                          *     ALL PREVIOUSLY SELECTED "MODES" AND RETURN POINTERS WILL BE CLEARED.
000806                          *     TO RE-EXECUTE A STRING OF LINKS PREVIOUSLY ENTERED, THE RESPONSE TO
000807                          *     "LINKS ?:" SHOULD BE A CARRIAGE-RETURN.
000808                          *     A STRING OF LINKS TERMINATED BY A C/R WITHOUT AN "H" OR "X" WILL
000809                          *     CONTINUOUSLY LOOP ON THAT STRING.
000810                          *
000811                          *
000812                          *     THE PROGRAM'S RESPONSE TO LINKS "A" OR "B" WILL HAVE THE
000813                          *     FOLLOWING FORMAT;
000814                          *
000815                          *        XXX--  DD DD DD DD DD DD DD DD DD DD DD DD DD DD DD DD
000816                          *
000817                          *     WHERE:
000818                          *          XXX = POSITION IN BUFFER OF FIRST DATA BYTE IN LINE (0-7F0 HEX)
000819                          *           DD = DATA BYTE FOUND IN BUFFER (0-FF)
000820                          *
000821                          *
000822                          *     THE RESPONSE TO LINK "P" IS AS FOLLOWS:
000823                          *
000824                          *        PARMS:   U LLL DD WWWW RRRR
000825                          *
000826                          *     WHERE:
000827                          *           U = UNIT NUMBER   (0-3)
000828                          *         LLL = RECORD LENGTH (1- 800 BYTES, HEX)
000829                          *          DD = DATA PATTERN  (0-FF)
000830                          *        WWWW = WRITE BUFFER LOCATION
000831                          *        RRRR = READ BUFFER LOCATION
000832                          *
000833                          *
000834                          *     THE RESPONSE TO LINK "Z" (PRINT STATUS, ETC.) IS AS FOLLOWS:
000835                          *
000836                          *        STAT:      U VVVV WWWW XXXX YYYY ZZZZ
000837                          *
000838                          *     WHERE:
000839                          *           U = UNIT (HEX)
000840                          *        VVVV = FILE (HEX)
000841                          *        WWWW = RECORD (HEX)
000842                          *        XXXX = RANGE REMAINING FROM CURRENT OPERATION (HEX)
000843                          *        YYYY = FIRST STATUS WORD
000844                          *        ZZZZ = SECOND STATUS WORD
000845                          *
000846                          *
000847                          *     SHOULD THE END-OF-TAPE MARKER BE DETECTED EITHER UPON
000848                          *     ENTERING OR WHILE OPERATION IN THE DEBUG MODE THE MESSAGE
000849                          *     "E-O-T" WILL BE REPORTED. THE PROGRAM WILL THEN HALT AND
000850                          *     WAIT FOR THE OPERATOR TO HIT "EXECUTE". THIS WILL CAUSE THE
000851                          *     PROGRAM TO CONTINUE OPERATION, BUT ALLOWS THE OPERATOR TO
000852                          *     TAKE APPROPRIATE MEASURES IF NECESSARY.
000853                          *
000854                          *     SHOULD LINK "D" DETECT ERRONEOUS DATA THE REPORT WILL FOLLOW THE
000855                          *     FORMAT LISTED UNDER "ERROR REPORTS".
000856                          *
000857                          *     LINKS "F" "G" AND "I" WILL FILL THE WRITE BUFFER WITH PRECISELY
000858                          *     THE NUMBER OF BYTES CALLED FOR IN THE "LENGTH" (RANGE)
000859                          *     PARAMETER. THE REMAINDER OF THE BUFFER WILL STAY UNCHANGED.
000860                          *
000861                          *     SELECTING LINKS "J" OR "K" WILL (IF MORE THAN 8K MEMORY IS
000862                          *     AVAILABLE) SET ASIDE A BUFFER RANDOMLY LOCATED ABOVE ZVSLR.
000863                          *     ONLY ONE RANDOM BUFFER IS ALLOWED AT A TIME. THEREFORE CHOOSING
000864                          *     A "J" FOLLOWED BY A "K" WILL NEGATE THE EFFECTS OF THE "J".
000865                          *
000866                          *     LINK "G" MAKES IT POSSIBLE TO CHECK A PREVIOUSLY RECORDED TAPE
000867                          *     BY FIRST READING THE DATA, TRANSFERRING THE DATA TO THE WRITE
000868                          *     BUFFER AND THEN RE-READING AND COMPARING THE "WRITE" AND "READ"
000869                          *     BUFFERS. FOR EXAMPLE:
000870                          *
000871                          *        LINKS ?: RGTRDH C/R
000872                          *
000873                          ************************************************************************
000874                          * MODE "E", USE
000875                          *     ENTERING MODE "E" SERVES TO ENABLE ERROR REPORTING, THUS
000876                          *     CANCELLING THE EFFECTS OF MODE "X". THE QUESTION "NEXT ?:"
000877                          *     IS THEN REPEATED IMMEDIATELY. "E" IS THE DEFAULT MODE WHEN
000878                          *     THE PROGRAM IS FRESHLY LOADED.
000879                          *
000880                          ************************************************************************
000881                          * MODE "F", USE
000882                          *     REQUESTING MODE "F" CAUSES THE FOLLOWING PRINTOUT:
000883                          *
000884                          *        NOW FFFF HEX FILE(S)... ?: NNNN C/R
000885                          *
000886                          *     WHERE;
000887                          *        FFFF = CURRENT NUMBER OF FILES TO BE USED IN
000888                          *               MODES A, G, R AND W (HEX).
000889                          *        NNNN = DESIRED NUMBER OF FILES (HEX).
000890                          *
000891                          *     NOTE: ZERO FILES WILL BE INTERPRETTED AS MEANING A REQUEST TO
000892                          *           GO TO END OF TAPE (E-O-T), AND THE REPORT WILL SAY:
000893                          *
000894                          *        E-O-T HEX FILE(S)... ?:
000895                          *
000896                          ************************************************************************
```

```
000897                          * MODE "I" USE
000898                          *     THE CHANNEL NUMBERS USED BY THE PROGRAM BECOME AVAILABLE
000899                          *     TO IT VIA THE RESPONSE TO THE QUESTION "CHANNEL ?:" WHEN
000900                          *     THE PROGRAM IS INITIALLY STARTED.  RESPONDING TO "NEXT ?:"
000901                          *     WITH "I" RETURNS THE PROGRAM TO THE CHANNEL QUESTION.
000902                          *     AFTER A HEX VALUE IS RECEIVED, THE PROGRAM WILL RECONFIGURE
000903                          *     THE CHANNEL ADDRESSES AND CHECK THAT A TAPE
000904                          *     SUBSYSTEM IS PRESENT ON THAT CHANNEL. IT WILL THEN RETURN
000905                          *     TO THE "NEXT ?:" QUESTION.
000906                          *
000907                          ************************************************************************
000908                          * MODE "P", USE
000909                          *     MODE "P" WILL PRINT A SUMMARY OF UP TO THE FIRST
000910                          *     TEN ERROR REPORTS RECEIVED IN THE PREVIOUS MODE(S). THE
000911                          *     REPORT WILL BE IN THE SAME FORMAT AS THE FIRST LINE OF
000912                          *     THE NORMAL ERROR REPORT, EXCEPT THAT THE DECIMAL PASS COUNT DURING
000913                          *     WHICH THE ERROR WAS RECORDED WILL ALSO BE SHOWN AT THE
000914                          *     END OF THE LINE.  DATA "IS-SB" ERRORS WILL NOT BE REPORTED.
000915                          *     THE ERROR SUMMARY TABLE WILL BE CLEARED UPON STARTING WITH
000916                          *     A CLEAN MODE.  (EG.  -0 -1 -2 -3). RUNNING MODE "P" WILL CLEAR
000917                          *     THE MODE TABLE.
000918                          *
000919                          ************************************************************************
000920                          * MODE "Q", USE
000921                          *     "QUICK" MODE WILL OPERATE IDENTICALLY TO MODE "A" AND ERRORS
000922                          *     WILL BE REPORTED WITH THE SAME REPORTING LABELS. HOWEVER,
000923                          *     INSTEAD OF TESTING FOR "N" DATA FILES, ONLY 16 DECIMAL (10 HEX)
000924                          *     DATA FILES WILL BE GENERATED AFTER COMPLETING THE PRELIMINARY
000925                          *     TESTS "T1" THROUGH "S1". THE PROGRAM WILL THEN REPORT "END-OF-
000926                          *     PASS" AND RESTART WITH TEST "AA".
000927                          *
000928                          ************************************************************************
000929                          * MODE "R", USE
000930                          *     THE "READ-ONLY" MODE IS INTENDED TO PROVIDE THE GREATEST
000931                          *     LIKELYHOOD OF SUCCESSFULLY READING A PRE-RECORDED TAPE AND
000932                          *     AS SUCH IT WILL OPERATE WITHOUT DOING ANY WRITE OR UNNECESSARY
000933                          *     SPACING OPERATIONS. IT WILL READ A TAPE PREVIOUSLY RECORDED
000934                          *     IN EITHER MODES "A", "J" OR "W" (SEE BELOW) UNTIL AN "END-OF-PASS
000935                          *     RECORD IS DETECTED WHEREUPON IT WILL REWIND AND START OVER.
000936                          *     THE "READ-ONLY" MODE PERFORMS AND REPORTS IDENTICALLY
000937                          *     TO MODE "A" WITH THE EXCEPTION THAT NO WRITE OPERATIONS
000938                          *     TAKE PLACE AND THAT A PASS IS CONSIDERED TO BE COMPLETE
000939                          *     WHEN AN "END-OF-PASS" RECORD IS DETECTED ON TAPE.
000940                          *
000941                          ************************************************************************
000942                          * MODE "S", USE
000943                          *     THE "STOP" MODE WILL HAVE THE EFFECT OF DE-SELECTING UNITS
000944                          *     0-3 AND THEN ASKING THE QUESTION "NEXT ?:".
000945                          *
000946                          ************************************************************************
000947                          * MODE "W", USE
000948                          *     THE "WRITE-ONLY" MODE PERFORMS AND REPORTS IDENTICALLY
000949                          *     TO MODE "A" WITH THE EXCEPTION THAT NO READING OR DATA
000950                          *     CHECKING OPERATIONS TAKE PLACE. "END-OF-PASS" IS
000951                          *     GENERATED WHEN EITHER "N" FILES HAVE BEEN WRITTEN OR
000952                          *     WHEN "END-OF-TAPE" HAS BEEN DETECTED. FOLLOWING "END-OF-PASS"
000953                          *     THE PROGRAM WILL "DE-SELECT" THAT DRIVE BUT CONTINUE TESTING
000954                          *     ANY OTHER UNITS. IF NO OTHER UNITS ARE SELECTED, THE "BREAK" KEY
000955                          *     MUST BE HIT TO RETURN TO "NEXT".
000956                          *
000957                          ************************************************************************
000958                          * MODE "X", USE
000959                          *     SELECTION OF MODE "X" SUPRESSES FUTURE ERROR REPORTS UNTIL
000960                          *     ENABLED BY MODE "E".  THE FIRST TEN ERROR REPORTS, IF
000961                          *     ANY, WILL CONTINUE TO BE ENTERED INTO THE ERROR HISTORY BUF-
000962                          *     FER.
000963                          *
000964                          ************************************************************************
000965                          * MODE "Z", USE
000966                          *     SELECTING MODE "Z" CAUSES THE QLT TEST TO BE RUN CONTINUOUSLY. THE
000967                          *     RESULTS OF THE QLT WILL BE INTERPRETED AND ANY ANOMALIES WILL BE
000968                          *     REPORTED. HITTING THE "BREAK" KEY WILL RETURN TO "NEXT" AFTER THE
000969                          *     CURRENT QLT HAS COMPLETED, AS MUCH AS 15 SECONDS.
000970                          *     DURING THE RUNNING OF MODE "Z" THE MESSAGE...
000971                          *
000972                          *        QLT PASS PP      (WHERE PP IS THE DECIMAL PASS COUNT)
000973                          *
000974                          *     WILL BE REPORTED AFTER EACH TEN DECIMAL PASSES OF THE QLT.
000975                          *
000976                          *     BEFORE RUNNING ANY SUBSEQUENT TESTS, MODE "S" SHOULD BE RUN.
000977                          *
000978                          ************************************************************************
000979                          * MODE "-", USE
000980                          *        THIS MODE WILL DELETE A SINGLE DRIVE FROM THE TEST LIST.
000981                          *
000982                          ************************************************************************
```

```
000983                                  /*****************************************************************
000984          0000                    ZERO    EQU     $
000985                                  XLOC    ZHISAZ,ZHRTCI,ZHRTCC,ZHRTCL,ZHNTSA,ZV$HR,ZHTH15,ZHCOMM,ZV$BKF
000986                                  XLOC    ZV$SVI,ZV$SV2,ZV$FRB,ZHIAFB,ZV$LR,ZV$ER,ZV$F,ZV$HRU,ZV$HRL
000987                                  *****************************************************************
000988          0100                    START   EQU     ZERO+X'100'
000989  0100                            ORG     START
000990                                  *
000991  0100  6280 2182                        LB      <GOFLAG,=Z'0001'
      0102  0001
000992  0103  0500 0229                        BBT     <RESTRT                 BYPASS THE FOLLOWING, DONE ALREADY
000993                                          CALL    ZV$RD,MSNAME            PRINT TITLE AND RESOURCES
      0105  FBC0 0003
      0107  D380 0000      X
      0109  0F80
      010A  21C1
000994  010B  8C51                            STS     =$R1                    GET STATUS
000995  010C  82D1                            LB      =$R1,=Z'2000'           CHECK 6/40 PRIV BIT
      010D  2000
000996  010E  050D                            BBT     >START2                 B IF 6/40, DON'T ASK "RTC FREQ ?"
000997                                          CALL    ZV$I,ZV$QC,MSRTC        PRINT RTC QUESTION
      010F  FBC0 0003
      0111  D380 0000      X
      0113  0F80
      0114  2284
000998                                          CALL    ZV$IH,ZV$ID,RTCHZ       GET RTC FREQUENCY (DEFAULT = 60)
      0115  FBC0 0003
      0117  D380 0000      X
      0119  0F80
      011A  21A0
000999                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
001000                                  * SET UP FOR NORMAL RUNNING AT LEVEL 15
001001                                  *
001002  011B  9B80 0126                START2  LAB     $B1,<LEVV15             CONTINUE AFTER SUSPEND TO LEV 15
001003  011D  9F80 230F                        STB     $B1,<SA15P
001004  011F  9B80 230C                        LAB     $B1,<SA15DV             NORMAL SAVE AREA
001005  0121  9F80 000F      X                 STB     $B1,<ZHISAZ+15*$AF      ISA15
001006  0123  8E70 800F                LEVXA   LEV     =Z'8000'+15             SUSPEND TO LEVEL 15
001007  0125  0000                            HLT                             LEV DIDN'T SUSPEND
001008                                  *
001009                                  * NOW AT LVL 15, SET UP FOR RTC RUPT TO LVL 5
001010                                  *
001011  0126  9B80 0FA6                LEVV15  LAB     $B1,<RTCFC
001012  0128  9F80 2303                        STB     $B1,<SA5P
001013  012A  9B80 2300                        LAB     $B1,<SA5DV
001014  012C  9F80 0005      X                 STB     $B1,<ZHISAZ+5*$AF
001015                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
001016                                  * CALIBRATE CPU CLOCK AGAINST RTC
001017                                  *
001018  012E  D380 0FA2                TI      LNJ     $B5,<LABEL              SET ERROR LABEL
001019  0130  5449                            DC      'TI'                    LABEL1
001020  0131  9800 21A0                        LDR     $R1,<RTCHZ              GET RTC FREQ
001021  0133  1F02                            MLV     $R1,=2
001022  0134  1E01                            ADV     $R1,=1
001023  0135  9F00 0000      X                 STR     $R1,<ZHRTCI             INITIAL VALUE OF RTC
001024  0137  9F00 0000      X                 STR     $R1,<ZHRTCC             CURRENT VALUE OF RTC
001025  0139  2C05                            LDV     $R2,=5
001026  013A  AF00 0000      X                 STR     $R2,<ZHRTCL             SET RTC TO RUPT AT LVL 5
001027  013C  D380 0FB3                        LNJ     $B5,<SYNCH              CALIBRATE
001028  013E  0005                RTCFA   RTCF                                    RTC DIDN'T RUPT
001029  013F  0000                            HLT                             RESTART
001030  0140  0F80 0100                        B       <START
001031                                  *****************************************************************
001032                                  * TIMING LOOP SHOULD NOW BE CALIBRATED
001033                                  *
001034  0142  8900 2182                RESUME  LBT     <GOFLAG,=Z'0001'        SET RESTART FLAG
      0144  0001
001035                                  *
001036                                  * NOW DO MODE I, CONFIGURE CHANNEL NUMBERS
001037                                  *
001038                                  MODEI   CALL    ZV$T,ZV$QC,MSCHAN       ASK QUESTION "CHANNEL ?:"
      0145  FBC0 0003
      0147  D380 0000      X
      0149  0F80
      014A  2221
001039                                          CALL    ZV$IH,CHAN              GET CHANNEL (DEFAULT = 1400)
      014B  FBC0 0003
      014D  D380 0000      X
      014F  0F80
      0150  22BA
001040  0151  D380 0FA2                MI      LNJ     $B5,<LABEL              ERROR LABEL
001041  0153  4D49                            DC      'MI'                    LABEL1
001042  0154  9800 22BA                        LDR     $R1,<CHAN               GET CHAN NO.
001043  0156  9B80 22BB                        LAB     $B1,<IOREAD             START OF CHANNELS TABLE
001044  0158  AB80 22D0                        LAB     $B2,<CHANZ              END OF CHANNELS TABLE
001045  015A  9AF1                MODEIA  SRM     $R1,+$B1,=Z'FE00'       CONFIGURE CHAN
      015B  FE00
001046  015C  9DD2                            CMB     $B1,=$B2                CHECK IF END OF TABLE
001047  015D  0200 015A                        BL      <MODEIA                 LOOP BACK, NOT DONE
001048                                  *
001049                                  * NOW CHECK THAT AT LEAST ONE DEVICE ADAPTER IS ON LINE BY
001050                                  * CHECKING ID CODES, BUT FIRST SET UP FOR NON-EXISTANT DEVICE
001051                                  * TRAP.
001052                                  *
001053  015F  9B80 2311                        LAB     $B1,<ISA1
001054  0161  9F80 0000      X                 STB     $B1,<ZHNTSA
001055  0163  9F80 2311                        STB     $B1,<ISA1               SO WE DON'T GET TSA FULL RUPT
001056  0165  8700 21B7                        CL      <TEMPA                  WILL SET IF GOOD ID FOUND
001057  0167  AB80 0203                        LAB     $B2,<MODEIF             TRAP HANDLER
001058  0169  AF80 0000      X                 STB     $B2,<ZHTH15             STORE IN TRAP 15 VECTOR
001059                                  *
001060  016B  8753                            CL      =$R3                    DRIVE INDEX AND COUNTER
001061  016C  A830 20F6                MODEIB  LDR     $R2,<DRIVE0,$R3         START WITH DRIVE 0
001062  016E  AA80 22C6                        SRM     $R2,<INIDEN,=Z'0180'    SET CHANNEL ADDRESS
      0170  0180
001063  0171  AA80 22C3                        SRM     $R2,<INBDC,=Z'0180'     SET CHANNEL ADDRESS
      0173  0180
001064  0174  AA80 22C5                        SRM     $R2,<INFIRM,=Z'0180'    SET CHANNEL ADDRESS
      0176  0180
001065  0177  AA80 22CF                        SRM     $R2,<INWAIT,=Z'0180'    SET CHANNEL ADDRESS
      0179  0180
001066  017A  AA80 22CD                        SRM     $R2,<INSTW2,=Z'0180'    SET CHANNEL ADDRESS
      017C  0180
```

```
001067  017D  8755                        CL      =$R5                    CLEAR IO RECEIVE ID CODE
001068  017E  8055                  MODEIC IO      =$R5,<INIDEN            GET ID
        017F  0000 22C6
001069  0181  07FD                        BIOF    >MODEIC
001070  0182  DF00 21B8                    STR     $R5,<TEMPB              SAVE ID FOR A WHILE
001071  0184  D570 FFE0                    AND     $R5,=Z'FFE0'            MASK IMPORTANT BITS
001072  0186  0970 2040                    CMR     $R5,=Z'2040'            IS IT A TAPE DRIVE ?
001073  0188  0980 01F1                    BNE     <MODEIE                 B IF NOT A TAPE
001074  018A  6900 21B7                    LBT     <TEMPA,=1               SET FLAG, TAPE IDENTIFIED
        018C  0001
001075  018D  0500 01A0                    BBT     <MODEID                 B IF FOLLOWING INFO ALREADY RCVD
001076  018F  8000 20E8            BDCIO   IO      <BDC,<INBDC             GET BDC TYPE (2 OR 3)
        0191  0000 22C3
001077  0193  07FC                        BIOF    >BDCIO
001078  0194  8052                  FIRMIO IO      =$R2,<INFIRM            GET FIRMWARE REV
        0195  0000 22C5
001079  0197  07FD                        BIOF    >FIRMIO
001080  0198  2048                        SOR     $R2,8                   RIGHT JUSTIFY IT
001081  0199  AF00 2180                    STR     $R2,<FIRM               SAVE IT FOR REPORT
001082  019B  8000 22B4            WTLP    IO      <WAITLP,<INWAIT         GET WAIT LOOP LOCATION
        019D  0000 22CF
001083  019F  07FC                        BIOF    >WTLP
001084                                    *
001085  01A0  8000 21B5            MODEID IO      <STAT2,<INSTW2          GET STATUS WORD 2
        01A2  0000 22CD
001086  01A4  07FC                        BIOF    >MODEID
001087  01A5  BA80 2228                    SRM     $R3,<MSDENS+3,=3        SET MESSAGE TO REPORT DRIVE NMBR
        01A7  0003
001088  01A8  8280 21B5                    LB      <STAT2,=Z'8000'         CHECK IF ON LINE
        01AA  8000
001089  01AB  0511                        BBT     >MODEIJ                 B IF ON LINE, OK
001090  01AC  8C80 2231                    LDI     <MSDOWN                 GET "OFF LINE" MESSAGE
001091  01AE  8D00 222C                    SDI     <MSDENS+7               SAVE IT FOR REPORT
001092  01B0  8C80 2233                    LDI     <MSDOWN+2
001093  01B2  8D00 222E                    SDI     <MSDENS+9
001094                                    CALL    ZV$T.ZV$TC,MSDENS       REPORT "OFF LINE"
        01B4  FBC0 0003
        01B6  D380 0000             X
        01B8  0F80
        01B9  2225
001095  01BA  0F80 01F1                    B       <MODEIE                 GO ON TO NEXT DRIVE
001096  01BC  8280 21B5            MODEIJ LB      <STAT2,=Z'1000'         CHECK FOR HI OR LO DENSITY
        01BE  1000
001097  01BF  059B                        BBF     >MODEIH                 B IF NRZI (IE. LOW DENSITY)
001098                                    *
001099                                    * APPEARS TO BE A P.E. DRIVE (HIGH DENSITY)
001100                                    *
001101  01C0  8C80 227B                    LDI     <MSPE                   GET PE MESSAGE
001102  01C2  8D00 222C                    SDI     <MSDENS+7               SAVE FOR REPORT
001103  01C4  8C80 227D                    LDI     <MSPE+2
001104  01C6  8D00 222E                    SDI     <MSDENS+9
001105                                    CALL    ZV$T.ZV$TC,MSDENS       REPORT DRIVE FINDINGS
        01C8  FBC0 0003
        01CA  D380 0000             X
        01CC  0F80
        01CD  2225
001106  01CE  8280 21B8                    LB      <TEMPB,=Z'0008'         CHECK PE BIT IN ID
        01D0  0008
001107  01D1  0504                        BBT     >MODEIG                 B IF STAT AND ID AGREE
001108  01D2  D380 1030                    LNJ     $B5,<FNER               STAT SAYS P.E., ID DOESN'T
001109  01D4  3035                  FNER05 DC      '05'                    LABEL2
001110  01D5  9870 1000            MODEIG LDR     $R1,=Z'1000'
001111  01D7  9F30 20FA                    STR     $R1,<DXDENS.$R3         SET DENSITY MASK
001112  01D9  0F98                        B       >MODEIE                 CHECK NEXT DRIVE IF ANY
001113                                    *
001114                                    * APPEARS TO BE AN NRZI DRIVE (LOW DENSITY)
001115                                    *
001116  01DA  8C80 2266            MODEIH LDI     <MSNRZI                 GET NRZI MESSAGE
001117  01DC  8D00 222C                    SDI     <MSDENS+7               SAVE FOR REPORT
001118  01DE  8C80 2268                    LDI     <MSNRZI+2
001119  01E0  8D00 222E                    SDI     <MSDENS+9
001120                                    CALL    ZV$T.ZV$TC,MSDENS       REPORT DRIVE FINDINGS
        01E2  FBC0 0003
        01E4  D380 0000             X
        01E6  0F80
        01E7  2225
001121  01E8  8280 21B8                    LB      <TEMPB,=Z'0004'         CHECK NRZI BIT IN ID
        01EA  0004
001122  01EB  0504                        BBT     >MODEII                 B IF STAT AND ID AGREE
001123  01EC  D380 1030                    LNJ     $B5,<FNER               STAT SAYS NRZI, ID DOESN'T
001124  01EE  3036                  FNER06 DC      '06'                    LABEL2
001125  01EF  8730 20FA            MODEII CL      <DXDENS.$R3             CLEAR DENSITY MASK
001126                                    *
001127  01F1  8AD3                  MODEIE INC     =$R3                    CHECK NEXT DRIVE
001128  01F2  3D03                        CMV     $R3,=3
001129  01F3  0380 016C                    BLE     <MODEIB                 GO BACK FOR NEXT DRIVE
001130                                    *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
001131                                    * ALL FOUR DEVICE ADDRESSES CHECKED, TEST THE FLAG
001132                                    *
001133  01F5  9B80 0F9A                    LAB     $B1,<TH15               SET UP FOR FUTURE...
001134  01F7  9F80 0000             X      STB     $B1,<ZHTH15             ...MISSING RESOURCE TRAPS.
001135  01F9  8280 21B7                    LB      <TEMPA,=Z'0001'
        01FB  0001
001136  01FC  0500 0208                    BBT     <GETBDC                 FLAG OK, CONTINUE
001137  01FE  D380 1030                    LNJ     $B5,<FNER               NO DEVICE ON LINE
001138  0200  4E44                  FNERND DC      'ND'                    LABEL2
001139  0201  0F80 0208                    B       <GETBDC
001140                                    *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
001141  0203  9B80 01F1            MODEIF LAB     $B1,<MODEIE             RESUME SCAN AT MODEIE
001142  0205  9F80 2317                    STB     $B1,<TSAIP
001143  0207  0003                        RTT                             RETURN FROM TRAP
001144                                    ********************************************************************************
001145                                    * REPORT CONTROLLER FIRMWARE LOAD AND REV NUMBER
001146                                    *
001147  0208  9800 20E8            GETBDC LDR     $R1,<BDC                LOOK AT BDC IDENTIFIER
001148  020A  9970 BDC3                    CMR     $R1,=Z'BDC3'
001149  020C  0908                        BE      >BDC3                   B IF BDC3, ELSE...
001150                                    *
001151                                    * FIRMWARE IS "BDC2"
001152                                    *
001153  020D  9870 4332                    LDR     $R1,=A'C2'              ASCII "C2"
001154  020F  9F00 2248                    STR     $R1,<MSFIRM+1           SET "BDC2" MESSAGE
001155  0211  8700 20E9                    CL      <BDCFLG                 CLEAR FLAG FOR BDC2
```

```
001156  0213  0F88                    B       >PRMSG              CONTINUE
001157                          *
001158                          * FIRMWARE IS "BDC3"
001159                          *
001160  0214  9870 4333          BDC3    LDR     $R1,=A'C3'          ASCII "C3"
001161  0216  9F00 2248                  STR     $R1,<MSFIRM+1       SET "BDC3" MESSAGE
001162  0218  8900 20E9                  LBT     <BDCFLG,=1          SET "BDC3" FLAG
        021A  0001
001163                          *
001164                          * NOW GET FIRMWARE REVISION NUMBER
001165                          *
001166                          PRMSG   CALL    ZV$T,ZV$TC,MSFIRM   PRINT "BDC- FIRMWARE REV"
        021B  FBC0 0003
        021D  D380 0000       X
        021F  0F80
        0220  2247
001167                                  CALL    ZV$TH,FIRM          PRINT THE FIRMWARE REV NUMBER
        0221  FBC0 0003
        0223  D380 0000       X
        0225  0F80
        0226  2180
001168  0227  D380 163D                LNJ     $B5,<QLTRUN         RUN QLT AND CHECK RESULTS
001169                          ********************************************************************
001170                          * SELECT DRIVES AND TESTS TO BE PERFORMED
001171                          *
001172  0229  D380 1634          RESTRT LNJ    $B5,<CLRMOD         CLEAR TO MODES -0 -1 -2 -3
001173                          *
001174  022B  9B80 2311                  LAB     $B1,<TSA1           FIRST TRAP SAVE AREA
001175  022D  9F80 0000       X          STB     $B1,<ZHNTSA
001176  022F  9B80 030B                  LAB     $B1,<X0             START OF SEQUENCER
001177  0231  9F80 102F                  STB     $B1,<SENSB5         RETURN ADDRESS FOR "SENSE" FIRST TIME
001178  0233  8700 2112                  CL      <EOTFLG             END-OF-TAPE FLAG
001179  0235  8700 20E7                  CL      <AUFL               TEST AU OPERATION FLAG
001180                          *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
001181                          * REPORT CURRENT MODES AND ASK "NEXT ?:"
001182                          *
001183  0237  0005              NEXT    RTCF                        STOP THE CLOCK
001184  0238  1C05                      LDV     $R1,=5
001185  0239  9F00 0000       X          STR     $R1,<ZHRTCL         SET RTC TO LVL 5
001186  023B  D380 1628                  LNJ     $B5,<CRLF           DO A LINE-FEED
001187  023D  2CFC                      LDV     $R2,=-4
001188  023E  9820 2106          NEXTA   LDR     $R1,<DXMD+4.$R2
001189  0240  D380 0FCB                  LNJ     $B5,<VDTR           PRINT 2 ASCII CHARS FROM R1 (MODE)
001190  0242  9870 2024                  LDR     $R1,='  $'
001191  0244  D380 0FCB                  LNJ     $B5,<VDTR           PRINT SINGLE SPACE
001192  0246  2780 023E                  BINC    $R2,<NEXTA          GO BACK FOR MORE
001193                                  CALL    ZV$T,ZV$G,MSNEXT    (PRINT QUESTION NEXT ?:)
        0248  FBC0 0003
        024A  D380 0000       X
        024C  0F80
        024D  2262
001194  024E  8700 219F                  CL      <RNGFLG             FLAG FOR ZERO RANGE IN DEBUG MODE
001195  0250  8700 2181                  CL      <FLAG               BIT 14=CR DETECTED, BIT 15=DRIVE MODIFIED
001196                          NEXTC   CALL    ZV$IA,ZVSTAT,TEMPA,DC8 (GET ASCII MODE)
        0252  FBC0 0003
        0254  D380 0000       X
        0256  0F80
        0257  21C0
        0258  21B7
        0259  20F1
001197  025A  9800 21C0                  LDR     $R1,<ZVSTAT         GET STATUS
001198  025C  8390 025E                  JMP     <NEXTB.$R1
001199  025E  0F7F              NEXTB   NOP     >$-1                STATUS ERROR
001200  025F  0F7F                      NOP     >$-1                BREAK
001201  0260  0FD7                      B       >NEXT               RUBOUT
001202  0261  0F85                      B       >NEXTE              COMMA
001203  0262  0F7F                      NOP     >$-1                C/R
001204                          *
001205  0263  8900 2181                  LBT     <FLAG,=Z'0002'      SET FLAG, CR DETECTED
        0265  0002
001206  0266  2C01              NEXTE   LDV     $R2,=1              SET INDEX FOR LDH TO FOLLOW
001207  0267  B0A0 21B7                  LDH     $R3,<TEMPA.$R2      GET RIGHT BYTE OF NEW SELECTION
001208  0269  3D30                      CMV     $R3,='0'            ASCII "0"
001209  026A  0203                      BL      >NEXTJ              B IF INVALID INPUT, CHECK FOR C/R
001210  026B  3D34                      CMV     $R3,='4'            ASCII "4"
001211  026C  0204                      BL      >NEXTI
001212  026D  3D0D              NEXTJ   CMV     $R3,=Z'0D'          CHECK FOR C/R
001213  026E  0980 02A4                  BNE     <NEXTK              B IF WASN'T C/R, INVALID INPUT
001214  0270  B570 0003          NEXTI   AND     $R3,=Z'0003'        MASK OFF UNUSED BITS
001215  0272  BF00 20EA                  STR     $R3,<BITE           DRIVE NUMBER AND TEMP INDEX
001216  0274  9800 21B7                  LDR     $R1,<TEMPA          GET MODE
001217  0276  1048                      SOR     $R1,8               RIGHT JUSTIFY IT
001218                          *
001219  0277  1D0D                      CMV     $R1,=X'0D'          C/R FOR NO CHANGE
001220  0278  0900 02ED                  BE      <NEXTF
001221  027A  1D45                      CMV     $R1,='E'            E FOR PRINT ERRORS AS THEY OCCUR
001222  027B  0900 0396                  BE      <MODEE
001223  027D  1D46                      CMV     $R1,='F'            F FOR FILES PER PASS
001224  027E  0900 0F7A                  BE      <MODEF
001225  0280  1D49                      CMV     $R1,='I'            I FOR IDENTIFICATION
001226  0281  0900 0145                  BE      <MODEI
001227  0283  1D4D                      CMV     $R1,='M'            M FOR MODIFY AND PATCH ROUTINE
001228  0284  0900 1745                  BE      <PCH
001229  0286  1D50                      CMV     $R1,='P'            P FOR PRINT ERROR TABLE
001230  0287  0900 03BD                  BE      <MODEP
001231  0289  1D53                      CMV     $R1,='S'            S FOR RESTART
001232  028A  0900 0100                  BE      <START
001233  028C  1D58                      CMV     $R1,='X'            X FOR SUPPRESS ERROR REPORTS
001234  028D  0900 039A                  BE      <MODEX
001235  028F  1D5A                      CMV     $R1,='Z'            Z FOR QLT TEST MODE
001236  0290  0900 039F                  BE      <MODEZ
001237  0292  1D41                      CMV     $R1,='A'            AN FOR ALL TESTS, DRIVE N
001238  0293  0900 02AC                  BE      <MODEAQ
001239  0295  1D44                      CMV     $R1,='D'            DN FOR DEBUG, DRIVE N
001240  0296  0900 0C3C                  BE      <MODED
001241  0298  1D51                      CMV     $R1,='Q'            QN FOR QUICK ALL, DRIVE N
001242  0299  0900 02AC                  BE      <MODEAQ
001243  029B  1D52                      CMV     $R1,='R'            RN FOR READ ONLY, DRIVE N
001244  029C  0900 02B0                  BE      <MODERW
001245  029E  1D57                      CMV     $R1,='W'            WN FOR WRITE ONLY, DRIVE N
001246  029F  0900 02B0                  BE      <MODERW
001247  02A1  1D2D                      CMV     $R1,='-'            -N FOR DELETE DRIVE N
001248  02A2  0900 02B4                  BE      <MZIP               DELETE
```

```
001249                                  NEXTK  CALL   ZV$I,ZV$TC,MSINVL    'INVALID MODE'
       02A4  FBC0 0003
       02A6  D380 0000      X
       02A8  0F80
       02A9  2250
001250 02AA  0F80 0237                         B      <NEXT               NO MATCH, TRY AGAIN
001251                                  ********************************************************************
001252                                  * MODES A Q R W OR DELETE (-N)
001253                                  *
001254 02AC  9B80 2323                  MODEAQ LAB    $B1,<SEQN           START OF SEQ TABLE FOR A, Q
001255 02AE  0F80 02B8                         B      <CMNA
001256                                  *
001257 02B0  9B80 232F                  MODERW LAB    $B1,<SEQ1           START OF SEQ TABLE FOR R, W
001258 02B2  0F80 02B8                         B      <CMNA
001259                                  *
001260 02B4  B800 20EA                  MZIP   LDR    $R3,<BITE           GET INDEX
001261 02B6  9CB0 210A                         LDB    $B1,<DXPT.$R3       GET OLD POINTER INTO SEQN TABLE
001262                                  *
001263 02B8  B800 20EA                  CMNA   LDR    $R3,<BITE           GET INDEX FOR MODER, MODEA
001264 02BA  9FB0 210A                         STB    $B1,<DXPT.$R3       SAVE NEW PNTR TO SEQN TABLE
001265                                  *
001266 02BC  9870 2D30                         LDR    $R1,='-0'           SET NEW MODE TO -N WHILE WE CHECK...
001267 02BE  9A53                              ADD    $R1,=$R3            IF OTHER MODES ARE NOT A Q R OR W...
001268 02BF  9F30 2102                         STR    $R1,<DXMD.$R3       THEN ERR TABLE WILL BE CLEARED, ETC.
001269                                  *
001270 02C1  9800 20EA                         LDR    $R1,<BITE           GET NEW DRIVE NUMBER
001271 02C3  1007                              SUL    $R1,7
001272 02C4  9900 20F5                         CMR    $R1,<DRIVE          CHECK AGAINST CURRENT DRIVE
001273 02C6  0980 02CB                         BNE    <CMNC               B IF CURRENT UNIT NOT BEING MODIFIED
001274 02C8  8900 2181                         LBT    <FLAG,=Z'0001'      SET FLAG, CURRENT UNIT MODIFIED
       02CA  0001
001275                                  *
001276                                  * CHECK IF PRIOR ERROR HISTORY SHOULD BE SAVED
001277                                  *
001278 02CB  3CFC                       CMNC   LDV    $R3,=-4
001279 02CC  9B80 2106                  CMND   LAB    $B1,<DXMD+4.$R3
001280 02CE  9281                              LLH    $R1,$B1             GET LEFT HALF OF "MODE"
001281 02CF  1D2D                              CMV    $R1,=X'2D'          "DASH"
001282 02D0  0990                              BNE    >CMNB               B IF ONE OF THE UNITS IS NOT A "DASH"
001283 02D1  37FB                              BINC   $R3,>CMND           TRY NEXT UNIT
001284                                  *
001285                                  * CLEAR ERROR TABLE AND COUNTER
001286                                  *
001287                                         CALL   ZV$F,ERTB,NULL,ERLNTH (BUF, PTRN, RNG)
       02D2  FBC0 0003
       02D4  D380 0000      X
       02D6  0F80
       02D7  2119
       02D8  218C
       02D9  2116
001288 02DA  8700 2113                         CL     <ERCT               TOTAL ERROR COUNT
001289 02DC  9B80 2119                         LAB    $B1,<ERTB
001290 02DE  9F80 217D                         STB    $B1,<ERTBPT         RUNNING POINTER THROUGH ERTB
001291                                  *
001292 02E0  B800 20EA                  CMNB   LDR    $R3,<BITE           GET NEW DRIVE NMBR
001293 02E2  9800 21B7                         LDR    $R1,<TEMPA          GET MODE
001294 02E4  9F30 2102                         STR    $R1,<DXMD.$R3       SAVE NEW MODE
001295 02E6  8730 20FE                         CL     <DXFN.$R3           CLEAR FILE COUNT
001296 02E8  8730 210E                         CL     <DXRN.$R3           CLEAR RECORD COUNT
001297 02EA  1C01                              LDV    $R1,=1
001298 02EB  9F30 2106                         STR    $R1,<DXPS.$R3       SET DRIVE PASS COUNT TO ONE
001299                                  *
001300 02ED  8280 2181                  NEXTF  LB     <FLAG,=Z'0002'      CHECK IF CR DETECTED
       02EF  0002
001301 02F0  0580 0252                         BBF    <NEXTC              B IF NO CR, GET MORE
001302 02F2  B800 2193                         LDR    $R3,<R3HOLD         RESTORE UNIT INDEX
001303 02F4  8280 2181                         LB     <FLAG,=Z'0001'      CHECK IF MODIFIED
       02F6  0001
001304 02F7  0504                              BBT    >NEXTG              B IF CURRENTLY RUNNING MODE MODIFIED
001305 02F8  DC80 102F                         LDB    $B5,<SENSB5
001306 02FA  8385                              JMP    $B5                 RETURN TO INTERRUPTING ROUTINE
001307
001308                                  * CURRENT TEST CANCELLED. CLEAR POSSIBLE DIAGNOSTIC OR ABNORMAL
001309                                  * CONFIGURATION MODES.
001310                                  *
001311 02FB  8000 22A9                  NEXTG  IO     <NORMAL,<OTCONF     (OUTPUT NORMAL CONFIGURATION)
       02FD  0000 22BD
001312 02FF  07FC                              BIOF   >NEXTG
001313 0300  D380 16EB                         LNJ    $B5,<INIZ           INITIALIZE MTC, ETC.
001314 0302  D380 033C                         LNJ    $B5,<MARW           ADVANCE SEQUENCE POINTERS, ETC.
001315 0304  9CB0 0307                         LDB    $B1,<NEXTH.$R3
001316 0306  8381                              JMP    $B1                 DO NEXT TEST IN SEQUENCE
001317                                  *
001318 0307  031F                       NEXTH  DC     <X1
001319 0308  0328                              DC     <X2
001320 0309  0331                              DC     <X3
001321 030A  030B                              DC     <X0
001322                                  ********************************************************************
001323                                  * DO TESTS, RESERVE B6 EXCLUSIVELY FOR THESE LNJ'S
001324                                  *
001325 030B  D380 100F                  X0     LNJ    $B5,<SENSE          SEE IF NEW MODE REQUESTED
001326 030D  2CFC                              LDV    $R2,=-4             COUNT 4 UNITS
001327 030E  9B80 2102                         LAB    $B1,<DXMD           MODE TABLE
001328 0310  90F1                       XOA    LDH    $R1,+$B1            GET MODE BYTE
001329 0311  1D2D                              CMV    $R1,='-'            IS IT A "DASH" ?
001330 0312  0984                              BNE    >XOB                B IF NOT DASH, MODE WAS SELECTED
001331 0313  27FB                              BINC   $R2,>XOA            TRY FOR ANOTHER UNIT
001332 0314  0F80 0237                         B      <NEXT               NOTHING SELECTED, GO TO "NEXT?"
001333 0316  3C00                       XOB    LDV    $R3,=0              SET INDEX
001334 0317  D380 0347                         LNJ    $B5,<COMX           CHECK MODE LIST, ACTIVE, REWIND, ETC.
001335 0319  0F86                              B      >X1                 SKIP THIS UNIT
001336                                  *
001337 031A  9CB0 210A                         LDB    $B1,<DXPT.$R3       B1 NOW POINTS TO SEQN TABLE
001338 031C  E389                              LNJ    $B6,*$B1            DO THE TEST
001339 031D  D380 033C                         LNJ    $B5,<MARW
001340                                  *
001341 031F  3C01                       X1     LDV    $R3,=1
001342 0320  D380 0347                         LNJ    $B5,<COMX
001343 0322  0F86                              B      >X2
001344                                  *
001345 0323  9CB0 210A                         LDB    $B1,<DXPT.$R3
001346 0325  E389                              LNJ    $B6,*$B1
001347 0326  D380 033C                         LNJ    $B5,<MARW
```

```
001348                                  *
001349  0328  3C02                      X2      LDV     $R3,=2
001350  0329  D380 0347                          LNJ     $B5,<COMX
001351  032B  0F86                              B       >X3
001352                                  *
001353  032C  9CB0 210A                          LDB     $B1,<DXPT.$R3
001354  032E  E389                              LNJ     $B6,*$B1
001355  032F  D380 033C                          LNJ     $B5,<MARW
001356                                  *
001357  0331  3C03                      X3      LDV     $R3,=3
001358  0332  D380 0347                          LNJ     $B5,<COMX
001359  0334  0FD7                              B       >X0
001360                                  *
001361  0335  9CB0 210A                          LDB     $B1,<DXPT.$R3
001362  0337  E389                              LNJ     $B6,*$B1
001363  0338  D380 033C                          LNJ     $B5,<MARW
001364  033A  0F80 030B                          B       <X0             GO TO TOP OF LOOP
001365                                  ********************************************************************
001366                                  * RESET MODE SEQUENCE POINTER IF END OF LIST
001367                                  *
001368  033C  9CB0 210A                  MARW    LDB     $B1,<DXPT.$R3
001369  033E  9BC1 0001                          LAB     $B1,$B1.$AF
001370  0340  8D89                              CMN     *$B1
001371  0341  0983                              BNE     >MARWA
001372                                  *
001373  0342  9B80 232F                          LAB     $B1,<SEQ1               START OF SEQ TABLE FOR DATA TESTS
001374  0344  9FB0 210A                  MARWA   STB     $B1,<DXPT.$R3
001375  0346  8385                              JMP     $B5                     RETURN
001376                                  ********************************************************************
001377                                  * DECIDE WHETHER OR NOT TO TEST NEXT UNIT
001378                                  *
001379  0347  DF80 0395                  COMX    STB     $B5,<COMXB5             SAVE B5 FOR RETURN
001380  0349  9830 2102                          LDR     $R1,<DXMD.$R3           GET MODE
001381  034B  1048                              SUR     $R1,8                   PUT "MODE" IN RIGHT BYTE
001382  034C  1D2D                              CMV     $R1,=X'2D'              IS IT A DASH?
001383  034D  0980 0350                          BNE     <COMXA                  B IF NOT A DASH, TEST DESIRED
001384  034F  8385                              JMP     $B5                     RETURN
001385                                  *
001386  0350  9C80 0005          X       COMXA   LDB     $B1,<ZHISAZ+5*$AF
001387  0352  AC80 000A          X               LDB     $B2,<ZHISAZ+10*$AF
001388  0354  BB80 036B                          LAB     $B3,<COMXC
001389  0356  BF80 230F                          STB     $B3,<SA15P
001390                                  *
001391  0358  CB80 0000          X               LAB     $B4,<ZHISAZ             START OF IV'S
001392  035A  DB80 003F          X               LAB     $B5,<ZHISAZ+63*$AF      END OF IV'S
001393  035C  8774                      COMXF   CL      +$B4                    CLEAR THE VECTOR
001394  035D  CDD5                              CMB     $B4,=$B5                DONE YET ?
001395  035E  03FE                              BLE     >COMXF                  B IF NOT DONE YET
001396  035F  CB80 230C                          LAB     $B4,<SA15DV             LVL 15 SAVE AREA
001397  0361  CF80 000F          X               STB     $B4,<ZHISAZ+15*$AF      SAVE IT IN VECTOR
001398                                  *
001399  0363  8C51                              STS     =$R1                    GET STATUS FOR CP RUPT LVL
001400  0364  9570 003F                          AND     $R1,=X'3F'
001401  0366  1D0F                              CMV     $R1,=15                 IS IT 15 ?
001402  0367  0904                              BE      >COMXC                  IF LVL ALREADY 15, DON'T SUSPEND
001403  0368  8E70 800F                  LEVXB   LEV     =Z'8000'+15
001404  036A  0000                              HLT                             NOT SET UP FOR A SECOND INTERRUPT
001405                                  *
001406                                  * NOW AT LEVEL 15, CONTINUE
001407                                  *
001408  036B  9F80 0005          X       COMXC   STB     $B1,<ZHISAZ+5*$AF
001409  036D  AF80 000A          X               STB     $B2,<ZHISAZ+10*$AF
001410  036F  9830 20F6                          LDR     $R1,<DRIVE0.$R3         GET DRIVE ADDRESS
001411  0371  9F00 20F5                          STR     $R1,<DRIVE              SAVE DRIVE ADDRESS
001412  0373  D380 1341                          LNJ     $B5,<SETCHN             SET UP CHANNEL ADDRESSES
001413  0375  9870 5830                          LDR     $R1,='X0'
001414  0377  9A53                              ADD     $R1,=$R3                ERROR LABEL
001415  0378  9F00 2186                          STR     $R1,<LABEL1             X0-X3
001416  037A  8000 21B4                          IO      <STAT1,<INSTW1          GET FIRST STATUS WORD
        037C  0000 22CC
001417  037E  0794                              BIOF    >COMXD                  SKIP THE TEST
001418  037F  8280 21B4                          LB      <STAT1,=Z'8000'         RDY ?
        0381  8000
001419  0382  050E                              BBT     >COMXB                  B IF RDY, DO THE TEST
001420  0383  8280 21B4                          LB      <STAT1,=Z'4000'         ATTN ?
        0385  4000
001421  0386  058C                              BBF     >COMXD                  IF N/RDY BUT N/ATTN, DON'T DO TEST
001422                                  *
001423                                  * GOT STATUS BUT ATTENTION WAS SET.
001424                                  * REPORT AN ERROR AND DESELECT DRIVE FROM FURTHER TESTING.
001425                                  *
001426  0387  9870 2D4E                          LDR     $R1,='-N'               GET AN ASCII "DASH"
001427  0389  9AB0 2102                          SRM     $R1,<DXMD.$R3,=Z'FF00'  DESELECT CURRENT DRIVE
        038B  FF00
001428  038C  D380 1030                          LNJ     $B5,<FNER               DEVICE N/RDY WITH ATTN BIT SET
001429  038E  3034                      FNER04  DC      '04'                    LABEL2
001430  038F  0F83                              B       >COMXD                  GO ON WITH NEXT TEST
001431                                  *
001432  0390  8A80 0395                  COMXB   INC     <COMXB5+$AF-1           INC B5 IN ORDER TO DO THE TEST
001433  0392  DC80 0395                  COMXD   LDB     $B5,<COMXB5             RESTORE B5
001434  0394  8385                              JMP     $B5                     RETURN
001435                                  *
001436  0395  0000                      COMXB5  RESV    $AF,0                   SAVE B5 HERE
001437                                  ********************************************************************
001438                                  * CLEAR FLAG TO REPORT ERRORS NORMALLY
001439                                  *
001440  0396  8700 21B6                  MODEE   CL      <SUPRES
001441  0398  0F80 0237                          B       <NEXT
001442                                  ********************************************************************
001443                                  * SET FLAG TO SUPPRESS ERROR REPORTS
001444                                  *
001445  039A  8900 21B6                  MODEX   LBT     <SUPRES,=Z'0001'
        039C  0001
001446  039D  0F80 0237                          B       <NEXT
001447                                  ********************************************************************
001448                                  * MODE Z, RUN QLT CONTINUOUSLY AND CHECK RESULTS
001449                                  *
001450  039F  8700 2106                  MODEZ   CL      <DXPS                   TOTAL PASS COUNTER
001451  03A1  1C0A                      MODEZA  LDV     $R1,=10
001452  03A2  9F00 2107                          STR     $R1,<DXPS+1             COUNT 10 PASSES PER REPORT
001453  03A4  D380 163D                  MODEZB  LNJ     $B5,<QLTRUN             DO QLT
001454  03A6  D380 100F                          LNJ     $B5,<SENSE              DO WE WANT "NEXT ?:"
001455  03A8  8A80 2106                          INC     <DXPS                   COUNT PASS
```

```
001456  03AA  8880 2107                    DEC    <DXPS+1              LOOP COUNT
001457  03AC  8980 2107                    CMZ    <DXPS+1
001458  03AE  0A76                         BAG    >MODEZB              B IF NOT DONE WITH PASS YET
001459                              *
001460                                     CALL   ZV$T1.ZV$TC,MSGLPS   REPORT PASS COUNT
        03AF  FBC0 0003
        03B1  D380 0000       X
        03B3  0F80
        03B4  227F
001461                                     CALL   ZV$TH.ZV$TD,DXPS
        03B5  FBC0 0003
        03B7  D380 0000       X
        03B9  0F80
        03BA  2106
001462  03BB  0F80 03A1                    B      <MODEZA              KEEP LOOPING
001463                              ********************************************************************
001464                              * MODE P, PRINT SUMMARY OF FIRST TEN (10) ERRORS.
001465                              *
001466  03BD  D380 1634             MODEP  LNJ    $B5,<CLRMOD          CLEAR TO MODES -0 -1 -2 -3
001467  03BF  9B80 2119                    LAB    $B1,<ERTB            START OF TABLE
001468                              *
001469  03C1  8900 218F             MPA    LBT    <PFLAG,=1            SET MODE "P" FLAG (FOR ERR REPORTS)
        03C3  0001
001470  03C4  9871                         LDR    $R1,+$B1
001471  03C5  1900 03FB                    BEZ    $R1,<MPDONE          B IF REST OF TABLE EMPTY
001472  03C7  9F00 2186                    STR    $R1,<LABEL1          LABEL1 ASCII
001473                              *
001474  03C9  9871                         LDR    $R1,+$B1
001475  03CA  9F00 2187                    STR    $R1,<LABEL2          LABEL2, ASCII
001476                              *
001477  03CC  ACF1                         LDB    $B2,+$B1
001478  03CD  AF80 107C                    STB    $B2,<FNERF           ERR LOC, HEX
001479                              *
001480  03CF  9871                         LDR    $R1,+$B1
001481  03D0  9F00 2114                    STR    $R1,<ERDRIV          DRIVE NUMBER, HEX
001482                              *
001483  03D2  9871                         LDR    $R1,+$B1
001484  03D3  9F00 2115                    STR    $R1,<ERFILE          FILE NUMBER, HEX
001485                              *
001486  03D5  9871                         LDR    $R1,+$B1
001487  03D6  9F00 2117                    STR    $R1,<ERREC           RECORD NUMBER, HEX
001488                              *
001489  03D8  BCF1                         LDB    $B3,+$B1
001490  03D9  BF80 21B3                    STB    $B3,<SLDB            DATA BUFFER ADDRESS, HEX
001491                              *
001492  03DB  9871                         LDR    $R1,+$B1
001493  03DC  9F00 21B4                    STR    $R1,<STAT1           STATUS WORD 1, HEX
001494                              *
001495  03DE  9871                         LDR    $R1,+$B1
001496  03DF  9F00 21B5                    STR    $R1,<STAT2           STATUS WORD 2, HEX
001497                              *
001498                              * NOW GO YE FORTH AND PRINT THE ERROR REPORT
001499                              *
001500  03E1  D380 106D                    LNJ    $B5,<FNERMP          REPORT THE ERROR SUMMARY
001501                              *
001502  03E3  9871                         LDR    $R1,+$B1
001503  03E4  9F00 21B7                    STR    $R1,<TEMPA
001504                                     CALL   ZV$TH.ZV$TD,TEMPA    PRINT PASS COUNT, DECIMAL
        03E6  FBC0 0003
        03E8  D380 0000       X
        03EA  0F80
        03EB  21B7
001505  03EC  8800 218F                    LBF    <PFLAG,=2            WAS BRK DETECTED? CLR FLAG
        03EE  0002
001506  03EF  0587                         BBF    >MPB                 B IF NO BRK FOUND, CONTINUE
001507  03F0  DB80 03F6                    LAB    $B5,<MPB
001508  03F2  DF80 102F                    STB    $B5,<SENSB5          RETURN HERE AFTER C/R AT "NEXT?"
001509  03F4  0F80 0237                    B      <NEXT                SERVICE THE BREAK
001510                              *
001511  03F6  AB80 217D             MPB    LAB    $B2,<ERTB+(8+2*$AF)*10 END OF BUFFER
001512  03F8  9DD2                         CMB    $B1,=$B2
001513  03F9  0200 03C1                    BL     <MPA                 LOOP BACK FOR MORE
001514  03FB  DB80 0237             MPDONE LAB    $B5,<NEXT           SET UP FOR RETURN
001515  03FD  DF80 102F                    STB    $B5,<SENSB5          C/R SHOULDN'T GET BACK IN-LINE
001516  03FF  8385                         JMP    $B5                  GO TO IT
001517                              ********************************************************************
001518                              *
001519                              * NEXT FOLLOWS MODE "A". FIRST COME THE TURN-AROUND AND OTHER
001520                              * DIAGNOSTIC TESTS. THEN COME THE NORMAL DATA TRANSFER TESTS.
001521                              *
001522                              ********************************************************************
```

```
001523                                  /*********************************************************************
001524                                  * TURN-AROUND TEST (DO ONLY ON "BDC2" FIRMWARE LOAD)
001525                                  *
001526                                  *     TRY THE DATA PATH. FIRST WRAP 1 BYTE OF
001527                                  *     DATA AROUND THE DAUGHTER-BOARD, THEN 2 BYTES
001528                                  *     (1 WORD), THEN 10 BYTES.  ZEROS + ONES ARE
001529                                  *     USED.  PATTERN OF 'AA55'
001530                                  *     BYTE-MODE IS THEN TRIED.
001531                                  *
001532                                  *       R1 = INDEX
001533                                  *       R4 = RANGE
001534                                  *       R6 = DATA
001535                                  *
001536  0400  8070 0000                 T1      IO      =0,<OTRUPT              INHIBIT INTERRUPTS
      0402  0000 22BF
001537  0404  8280 20E9                         LB      <BDCFLG,=1              CHECK FIRMWARE
      0406  0001
001538  0407  0583                             BBF     >T1A                    B IF "BDC2" FIRMWARE
001539  0408  0F80 04E3                         B       <T9A                    SKIP T1-T9 IF "BDC3" FIRMWARE
001540                                  *
001541  040A  D380 0FA2                 T1A     LNJ     $B5,<LABEL
001542  040C  5431                             DC      'T1'                    LABEL1
001543  040D  D380 16EB                         LNJ     $B5,<INIZ               INITIALIZE SUBSYSTEM
001544  040F  8000 22A0                 T1B     IO      <ERASE,<OTTASK          MUST BE OFF BUT FOR T1-T9
      0411  0000 22C2
001545  0413  07FC                             BIOF    >T1B
001546  0414  D380 136E                         LNJ     $B5,<GETRNG             WAIT FOR RDY
001547                                  *
001548  0416  4C01                             LDV     $R4,=1                  RANGE
001549  0417  CF00 219E                         STR     $R4,<RNG                RANGE
001550  0419  D380 1569                         LNJ     $B5,<GOWRAP             RETRY ROUTINE, IF NEEDED, IS A NOP
001551                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
001552                                  *
001553  041B  D380 0FA2                 T2      LNJ     $B5,<LABEL
001554  041D  5432                             DC      'T2'                    LABEL1
001555  041E  4C02                             LDV     $R4,=2
001556  041F  CF00 219E                         STR     $R4,<RNG                RANGE
001557  0421  D380 1569                         LNJ     $B5,<GOWRAP
001558                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
001559                                  *
001560  0423  D380 0FA2                 T3      LNJ     $B5,<LABEL
001561  0425  5433                             DC      'T3'                    LABEL1
001562  0426  4C0F                             LDV     $R4,=X'F'
001563  0427  CF00 219E                         STR     $R4,<RNG                RANGE
001564  0429  D380 1569                         LNJ     $B5,<GOWRAP
001565                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
001566                                  *
001567  042B  D380 0FA2                 T4      LNJ     $B5,<LABEL
001568  042D  5434                             DC      'T4'                    LABEL1
001569  042E  4C10                             LDV     $R4,=X'10'
001570  042F  CF00 219E                         STR     $R4,<RNG                RANGE
001571  0431  D380 1569                         LNJ     $B5,<GOWRAP
001572                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
001573                                  *          WRAP  AA55 55AA AA55 55AA  DATA PATTERN
001574                                  *
001575  0433  D380 0FA2                 T5      LNJ     $B5,<LABEL
001576  0435  5435                             DC      'T5'                    LABEL1
001577  0436  E870 AA55                         LDR     $R6,=Z'AA55'            SPECIAL PATTERN
001578  0438  D380 1591                         LNJ     $B5,<LDBUF
001579  043A  6018                             SCL     $R6,8
001580  043B  1C01                             LDV     $R1,=1
001581  043C  EF10 17E6                 T5A     STR     $R6,<WBUF.$R1
001582  043E  1E02                             ADV     $R1,=2
001583  043F  1D09                             CMV     $R1,=9
001584  0440  027C                             BL      >T5A
001585  0441  8700 218E                         CL      <OFSETW                 WRITE OFFSET
001586  0443  8700 218D                         CL      <OFSETR                 READ OFFSET
001587  0445  D380 1599                         LNJ     $B5,<TURN
001588  0447  D380 10B6                         LNJ     $B5,<SET                SET UP FOR RETRY MESSAGE
001589  0449  9B80 044F                         LAB     $B1,<T6                 RETRY LOCATION
001590  044B  9F80 21A4                         STB     $B1,<RTRY
001591  044D  D380 15F9                         LNJ     $B5,<COMPAR
001592                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
001593                                  * BYTE-CONFIGURATION WRAP-AROUND (ODD OUT - ODD IN)
001594                                  *
001595  044F  D380 0FA2                 T6      LNJ     $B5,<LABEL
001596  0451  5436                             DC      'T6'                    LABEL1
001597  0452  9B80 17E6                         LAB     $B1,<WBUF
001598  0454  4C12                             LDV     $R4,=X'12'
001599  0455  CF00 219E                         STR     $R4,<RNG                RANGE
001600  0457  E870 0102                         LDR     $R7,=X'0102'            DATA PATTERN
001601  0459  D380 160F                         LNJ     $B5,<INCBFR             INCREMENTING DATA
001602  045B  1C01                             LDV     $R1,=1                  BYTE-CONFIGURATION INDEX
001603  045C  9F00 218E                         STR     $R1,<OFSETW             WRITE OFFSET
001604  045E  9F00 218D                         STR     $R1,<OFSETR             READ OFFSET
001605  0460  4C10                             LDV     $R4,=X'10'
001606  0461  CF00 219E                         STR     $R4,<RNG                RANGE
001607  0463  D380 1599                         LNJ     $B5,<TURN
001608  0465  E870 3302                         LDR     $R6,=X'3302'
001609  0467  EF00 17E6                         STR     $R6,<WBUF
001610  0469  4C11                             LDV     $R4,=X'11'              COMPARE RANGE (BYTES)
001611  046A  CF00 219E                         STR     $R4,<RNG                RANGE
001612  046C  D380 10B6                         LNJ     $B5,<SET                SET UP OR RETRY MESSAGE
001613  046E  9B80 0474                         LAB     $B1,<T7                 RETRY LOCATION
001614  0470  9F80 21A4                         STB     $B1,<RTRY
001615  0472  D380 15F9                         LNJ     $B5,<COMPAR
001616                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
001617                                  *     BYTE-CONFIGURATION  (ODD OUT - EVEN IN)
001618                                  *
001619  0474  D380 0FA2                 T7      LNJ     $B5,<LABEL
001620  0476  5437                             DC      'T7'                    LABEL1
001621  0477  9B80 17E6                         LAB     $B1,<WBUF
001622  0479  4C12                             LDV     $R4,=X'12'              RANGE
001623  047A  CF00 219E                         STR     $R4,<RNG                RANGE
001624  047C  7C01                             LDV     $R7,=1                  DATA PATTERN
001625  047D  D380 160F                         LNJ     $B5,<INCBFR
001626  047F  1C01                             LDV     $R1,=1
001627  0480  9F00 218E                         STR     $R1,<OFSETW             WRITE OFFSET
001628  0482  8700 218D                         CL      <OFSETR                 READ OFFSET
001629  0484  4C10                             LDV     $R4,=X'10'
001630  0485  CF00 219E                         STR     $R4,<RNG                RANGE
001631  0487  D380 1599                         LNJ     $B5,<TURN
001632  0489  9B80 17E6                         LAB     $B1,<WBUF
```

```
001633  048B  4C12                    LDV     $R4,=X'12'
001634  048C  CF00 219E                STR     $R4,<RNG                RANGE
001635  048E  F870 0102                LDR     $R7,=X'0102'            DATA PATTERN
001636  0490  D380 160F                LNJ     $B5,<INCBFR
001637  0492  8880 219E                DEC     <RNG
001638  0494  8880 219E                DEC     <RNG                    RANGE =X'0010'
001639  0496  D380 10B6                LNJ     $B5,<SET                SET UP FOR RETRY MESSAGE
001640  0498  9B80 049E                LAB     $B1,<T8                 RETRY LOCATION
001641  049A  9F80 21A4                STB     $B1,<RTRY
001642  049C  D380 15F9                LNJ     $B5,<COMPAR
001643                          *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
001644                          *     BYTE-CONFIGURATION  (EVEN OUT - ODD IN)
001645                          *
001646  049E  D380 0FA2         T8      LNJ     $B5,<LABEL
001647  04A0  5438                      DC      'T8'                    LABEL1
001648  04A1  4C12                      LDV     $R4,=X'12'
001649  04A2  CF00 219E                 STR     $R4,<RNG                RANGE
001650  04A4  F870 0102                 LDR     $R7,=X'0102'
001651  04A6  9B80 17E6                 LAB     $B1,<WBUF
001652  04A8  D380 160F                 LNJ     $B5,<INCBFR
001653  04AA  8700 218E                 CL      <OFSETW                 WRITE OFFSET
001654  04AC  1C01                      LDV     $R1,=1                  READ OFFSET
001655  04AD  9F00 218D                 STR     $R1,<OFSETR
001656  04AF  4C10                      LDV     $R4,=X'10'
001657  04B0  CF00 219E                 STR     $R4,<RNG                RANGE
001658  04B2  D380 1599                 LNJ     $B5,<TURN
001659  04B4  9B80 17E6                 LAB     $B1,<WBUF
001660  04B6  4C11                      LDV     $R4,=X'11'
001661  04B7  CF00 219E                 STR     $R4,<RNG                RANGE
001662  04B9  7C01                      LDV     $R7,=1                  DATA PATTERN
001663  04BA  D380 160F                 LNJ     $B5,<INCBFR
001664  04BC  F870 3301                 LDR     $R7,=X'3301'
001665  04BE  FF00 17E6                 STR     $R7,<WBUF
001666  04C0  D380 10B6                 LNJ     $B5,<SET                SET UP FOR RETRY MESSAGE
001667  04C2  9B80 04C8                 LAB     $B1,<T9                 RETRY LOCATION
001668  04C4  9F80 21A4                 STB     $B1,<RTRY
001669  04C6  D380 15F9                 LNJ     $B5,<COMPAR
001670                          *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
001671                          * WRAP DAUGHTER-BOARD WITH RANDOM DATA
001672                          *
001673  04C8  D380 0FA2         T9      LNJ     $B5,<LABEL
001674  04CA  5439                      DC      'T9'                    LABEL1
001675  04CB  4C10                      LDV     $R4,=X'10'
001676  04CC  CF00 219E                 STR     $R4,<RNG
001677                                  CALL    ZV$FR,WBUF,DC8          (RANDOM DATA, 8 WORDS)
        04CE  FBC0 0003
        04D0  D380 0000         X
        04D2  0F80
        04D3  17E6
        04D4  20F1
001678  04D5  8700 218E                 CL      <OFSETW                 WRITE OFFSET
001679  04D7  8700 218D                 CL      <OFSETR                 READ OFFSET
001680  04D9  D380 1599                 LNJ     $B5,<TURN
001681  04DB  D380 10B6                 LNJ     $B5,<SET                SET UP FOR RETRY MESSAGE
001682  04DD  9B80 04E3                 LAB     $B1,<T9A                RETRY LOCATION
001683  04DF  9F80 21A4                 STB     $B1,<RTRY
001684  04E1  D380 15F9                 LNJ     $B5,<COMPAR
001685  04E3  8000 22A9         T9A     IO      <NORMAL,<OTCONF         GET OUT OF DIAGNOSTIC CONFIGURATION
        04E5  0000 22BD
001686  04E7  07FC                      BIOF    >T9A
001687  04E8  8386                      JMP     $B6                     RETURN TO HANDLER
001688                          ********************************************************************************
001689                          * REWIND TEST. GET TO B-O-T, PARTIALLY CHECK REWIND OPERATION
001690                          *
001691  04E9  D380 0FA2         RW      LNJ     $B5,<LABEL
001692  04EB  5257                      DC      'RW'                    LABEL1
001693  04EC  D380 0FEC                 LNJ     $B5,<INSTAT             GET STATUS
001694  04EE  8280 21B4                 LB      <STAT1,=Z'0200'         CHECK FOR BOT
        04F0  0200
001695  04F1  0500 0531                 BBT     <RWE                    B IF ALREADY AT BOT
001696                          *
001697                          * REWIND TO BOT
001698                          *
001699  04F3  8000 2299                 IO      <REWIND,<OTTASK         REWIND
        04F5  0000 22C2
001700  04F7  0704                      BIOT    >RWA                    B IF COMMAND TAKEN
001701  04F8  D380 1030                 LNJ     $B5,<FNER               SHOULD HAVE BEEN READY FOR COMMAND
001702  04FA  3130                      DC      '10'                    LABEL2
001703                          *
001704  04FB  D380 0FC5         RWA     LNJ     $B5,<TIME               TIMEOUT FOR...
001705  04FD  0064                      DC      100                     100 MILLISECONDS
001706  04FE  8000 22B2                 IO      <STOPIO,<OTCONT         RESET CHAN BSY FLOP
        0500  0000 22BE
001707  0502  D380 0FEC         RWAA    LNJ     $B5,<INSTAT             GET STATUS
001708  0504  8280 21B5                 LB      <STAT2,=Z'4000'         WAIT FOR REWIND TO START
        0506  4000
001709  0507  05FB                      BBF     >RWAA                   B IF NOT REWINDING YET
001710                          *
001711  0508  D380 0FEC                 LNJ     $B5,<INSTAT             GET STAT
001712  050A  8980 21B4                 CMZ     <STAT1
001713  050C  0988                      BNE     >RWB                    B IF STAT1 NG
001714  050D  9800 21B5                 LDR     $R1,<STAT2
001715  050F  9630 20FA                 XOR     $R1,<DXDENS,$R3         MASK WITH DENSITY BIT
001716  0511  9670 C000                 XOR     $R1,=Z'C000'            SB ON-LINE, RWND
001717  0513  1904                      BEZ     $R1,>RWC
001716  0514  D380 1030         RWB     LNJ     $B5,<FNER               STAT NG DURING RWND
001719  0516  3131                      DC      '11'                    LABEL2
001720                          *
001721                          * NOW WAIT FOR RDY WHILE REWINDING
001722                          *
001723  0517  D380 0FEC         RWC     LNJ     $B5,<INSTAT             GET STATUS
001724  0519  8280 21B4                 LB      <STAT1,=Z'8000'         WAIT FOR RDY
        051B  8000
001725  051C  05FB                      BBF     >RWC                    B IF N/RDY YET (WAIT)
001726  051D  8000 21B5         RWCC    IO      <STAT2,<INSTW2          GET STATUS WORD 2
        051F  0000 22CD
001727  0521  07FC                      BIOF    >RWCC
001728  0522  9800 21B4                 LDR     $R1,<STAT1              RDY NOW, CHK FOR VALID STATUS
001729  0524  9670 C200                 XOR     $R1,=Z'C200'            SB RDY, ATTN, BOT
001730  0526  1988                      BNEZ    $R1,>RWD                B IF NG
001731  0527  9800 21B5                 LDR     $R1,<STAT2
001732  0529  9630 20FA                 XOR     $R1,<DXDENS,$R3         MASK WITH DENSITY BIT
001733  052B  9670 8000                 XOR     $R1,=Z'8000'            SB BOT (ON LINE)
```

```
001734  052D  1904           BEZ   $R1,>RWE              B IF OK
001735  052E  D380 1030  RWD  LNJ   $B5,<FNER             STAT NG AFTER REWIND
001736  0530  3132      FNER12 DC   '12'                  LABEL2
001737                   *
001738                   * UNIT AT B-O-T, DO ANOTHER REWIND, SHOULD HAVE NO EFFECT
001739                   *
001740  0531  8000 2299  RWE  IO    <REWIND,<OTTASK       ISSUE REWIND
        0533  0000 22C2
001741  0535  07FC           BIOF  >RWE                  HANG IN THERE
001742  0536  0F7F           NOP   >$-1                  DELAY
001743  0537  0F7F           NOP   >$-1                  DELAY SOME MORE
001744  0538  8000 21B4  RWF  IO    <STAT1,<INSTW1        GET STAT1
        053A  0000 22CC
001745  053C  07FC           BIOF  >RWF                  CAN'T GET STATUS WORD ONE
001746  053D  9800 21B4      LDR   $R1,<STAT1
001747  053F  9970 8200      CMR   $R1,=Z'8200'          RDY, BOT
001748  0541  0980           BNE   >RWN                  B IF NG
001749  0542  8000 21B5  RWFF IO    <STAT2,<INSTW2        GET SECOND STATUS WORD
        0544  0000 22CD
001750  0546  07FC           BIOF  >RWFF                 B IF CANT GET IT
001751  0547  9800 21B5      LDR   $R1,<STAT2
001752  0549  9630 20FA      XOR   $R1,<DXDENS,$R3       MASK WITH DENSITY BIT
001753  054B  9970 8000      CMR   $R1,=Z'8000'          ON-LINE
001754  054D  0904           BE    >RWH                  B IF OK
001755  054E  D380 1030  RWG  LNJ   $B5,<FNER             CANT GET STAT OR NG AFTER RWND FROM BOT
001756  0550  3133      FNER13 DC   '13'                  LABEL2
001757                   *
001758  0551  8386      RWH  JMP   $B6                    END OF TEST Rw, RETURN TO HANDLER
001759                   ************************************************************
001760                   * ERASE TAPE, TIME ERASE FROM BOT AND CHECK FOR TAPE MOTION OFF BOT.
001761                   *
001762  0552  D380 0FA2  ER   LNJ   $B5,<LABEL
001763  0554  4552           DC    'ER'                  LABEL1
001764  0555  8000 22A0      IO    <ERASE,<OTTASK        ERASE
        0557  0000 22C2
001765  0559  07F9           BIOF  >ER
001766                   *
001767  055A  8700 21BA      CL    <TIMCNT               SET UP FOR TIMING
001768  055C  D380 0FC5  ERA  LNJ   $B5,<TIME             TIMEOUT FOR...
001769  055E  0001           DC    1                     1 MS
001770  055F  8000 21B4      IO    <STAT1,<INSTW1        TRY TO GET STATUS
        0561  0000 22CC
001771  0563  8A80 21BA      INC   <TIMCNT               ACCUMULATE TIMER
001772  0565  07F7           BIOF  >ERA                  B IF NOT DONE
001773  0566  8000 21B5      IO    <STAT2,<INSTW2
        0568  0000 22CD
001774                   *
001775  056A  9800 21B4      LDR   $R1,<STAT1
001776  056C  9970 8000      CMR   $R1,=Z'8000'
001777  056E  0988           BNE   >ERB                  B IF STAT 1 NG
001778  056F  9800 21B5      LDR   $R1,<STAT2
001779  0571  9630 20FA      XOR   $R1,<DXDENS,$R3       MASK WITH DENSITY BIT
001780  0573  9970 8000      CMR   $R1,=Z'8000'
001781  0575  0904           BE    >ERC                  B IF STAT OK
001782  0576  D380 1030  ERB  LNJ   $B5,<FNER             STATUS NG AFTER ERASE
001783  0578  3139      FNER19 DC   '19'                  LABEL2
001784                   *
001785                   * NOW CHECK AGAINST ALLOWED TIME LIMITS.
001786                   *
001787  0579  D380 1709  ERC  LNJ   $B5,<TIMCHK           CHECK NOMINAL VALUE, ERR IF OUT OF LIMITS.
001788  057B  2321           DC    <TIMER                CONTAINS NOMINAL TIME
001789                   *
001790  057C  D380 100F      LNJ   $B5,<SENSE            NEXT?
001791  057E  8386           JMP   $B6                   RETURN
001792                   ************************************************************
001793                   * FILE MARK TEST. WRITE FM FOLLOWING AN ERASE (IE., NOT FROM BOT).
001794                   * TIME THE OPERATION AND CHECK STATUS.
001795                   *
001796  057F  D380 0FA2  FM   LNJ   $B5,<LABEL
001797  0581  464D           DC    'FM'                  LABEL1
001798  0582  8000 22A1      IO    <WRITFM,<OTTASK       WRITE FILE MARK
        0584  0000 22C2
001799  0586  07F9           BIOF  >FM
001800                   *
001801  0587  8700 21BA      CL    <TIMCNT               SET UP FOR TIMING
001802  0589  D380 0FC5  FMA  LNJ   $B5,<TIME             TIMEOUT FOR...
001803  058B  0001           DC    1                     1 MS
001804  058C  8000 21B4      IO    <STAT1,<INSTW1        TRY TO GET STATUS
        058E  0000 22CC
001805  0590  8A80 21BA      INC   <TIMCNT               ACCUMULATE TIME
001806  0592  07F7           BIOF  >FMA                  B IF NOT DONE
001807  0593  8000 21B5      IO    <STAT2,<INSTW2        GET STAT
        0595  0000 22CD
001808                   *
001809  0597  9800 21B4      LDR   $R1,<STAT1
001810  0599  9970 8400      CMR   $R1,=Z'8400'          RDY, FM DET
001811  059B  0988           BNE   >FMB                  B IF STAT1 NG
001812  059C  9800 21B5      LDR   $R1,<STAT2
001813  059E  9630 20FA      XOR   $R1,<DXDENS,$R3       MASK WITH DENSITY BIT
001814  05A0  9970 8000      CMR   $R1,=Z'8000'          ON LINE
001815  05A2  0904           BE    >FMC                  B IF STAT OK
001816  05A3  D380 1030  FMB  LNJ   $B5,<FNER             STAT NG AFTER "WRITE FILE"
001817  05A5  3231      FNER21 DC   '21'                  LABEL2
001818                   *
001819  05A6  D380 1709  FMC  LNJ   $B5,<TIMCHK           CHECK TIME, ERROR IF NG
001820  05A8  2322           DC    <TIMFM                NOMINAL TIME FOR WFM, N/BOT, 45 IPS
001821                   *
001822  05A9  D380 100F      LNJ   $B5,<SENSE            NEXT?
001823  05AB  8386           JMP   $B6                   RETURN
001824                   ************************************************************
001825                   * BACKSPACE A RECORD, (IS A FM). CHECK THAT DATA WAS WRITTEN IN TEST 'FM'.
001826                   *
001827  05AC  D380 0FA2  BR   LNJ   $B5,<LABEL
001828  05AE  4252           DC    'BR'                  LABEL1
001829  05AF  D380 1453      LNJ   $B5,<BSR              START TO BACK SPACE A REC
001830  05B1  D380 0FEC      LNJ   $B5,<INSTAT           WAIT FOR RDY AND GET STAT
001831  05B3  9800 21B4      LDR   $R1,<STAT1
001832  05B5  9970 8400      CMR   $R1,=Z'8400'          RDY, FM DET, N/BOT, ETC.
001833  05B7  0988           BNE   >BRA                  B IF STAT1 NG
001834  05B8  9800 21B5      LDR   $R1,<STAT2
001835  05BA  9630 20FA      XOR   $R1,<DXDENS,$R3       MASK WITH DENSITY BIT
001836  05BC  9970 8000      CMR   $R1,=Z'8000'          ON LINE
001837  05BE  0904           BE    >BRB                  B IF STAT OK
```

```
001838  05BF  D380 1030          BRA     LNJ     $B5,<FNER               STAT NG AFTER 'BSR'
001839  05C1  3235               FNER25  DC      '25'                    LABEL2
001840                           *
001841  05C2  D380 100F          BRB     LNJ     $B5,<SENSE              NEXT?
001842  05C4  8386                      JMP     $B6                     RETURN
001843                           ********************************************************************
001844                           * FORWARD SPACE A RECORD AND CHECK STATUS. (REC IS A FM)
001845                           *
001846  05C5  D380 0FA2          FR      LNJ     $B5,<LABEL
001847  05C7  4652                       DC      'FR'                    LABEL1
001848  05C8  D380 13E9                  LNJ     $B5,<FSR                START IO SPACE FORWARD
001849  05CA  D380 0FEC                  LNJ     $B5,<INSTAT             GET STATUS
001850  05CC  9800 21B4                  LDR     $R1,<STAT1
001851  05CE  9970 8400                  CMR     $R1,=Z'8400'            RDY, FM DET
001852  05D0  0988                       BNE     >FRA                    B IF STAT1 NG
001853  05D1  9800 21B5                  LDR     $R1,<STAT2
001854  05D3  9630 20FA                  XOR     $R1,<DXDENS,$R3         MASK WITH DENSITY BIT
001855  05D5  9970 8000                  CMR     $R1,=Z'8000'            ON LINE
001856  05D7  0904                       BE      >FRB                    B IF STAT OK
001857  05D8  D380 1030          FRA     LNJ     $B5,<FNER               STAT NG AFTER FSR WHICH IS FM
001858  05DA  3239               FNER29  DC      '29'                    LABEL2
001859                           *
001860  05DB  D380 100F          FRB     LNJ     $B5,<SENSE              NEXT?
001861  05DD  8386                      JMP     $B6                     RETURN
001862                           ********************************************************************
001863                           * BACKSPACE A FILE AND CHECK STATUS
001864                           *
001865  05DE  D380 0FA2          BF      LNJ     $B5,<LABEL
001866  05E0  4246                       DC      'BF'                    LABEL1
001867  05E1  D380 1414                  LNJ     $B5,<BSF                START IO BACKSPACE A FILE
001868  05E3  D380 0FEC                  LNJ     $B5,<INSTAT             GET STATUS
001869  05E5  9800 21B4                  LDR     $R1,<STAT1
001870  05E7  9970 8400                  CMR     $R1,=Z'8400'            RDY, FM DET
001871  05E9  0988                       BNE     >BFA                    B IF STAT1 NG
001872  05EA  9800 21B5                  LDR     $R1,<STAT2
001873  05EC  9630 20FA                  XOR     $R1,<DXDENS,$R3         MASK WITH DENSITY BIT
001874  05EE  9970 8000                  CMR     $R1,=Z'8000'            ON LINE
001875  05F0  0904                       BE      >BFB                    B IF STAT OK
001876  05F1  D380 1030          BFA     LNJ     $B5,<FNER               STAT NG AFTER BSF
001877  05F3  3330               FNER30  DC      '30'                    LABEL2
001878                           *
001879  05F4  D380 100F          BFB     LNJ     $B5,<SENSE              NEXT?
001880  05F6  8386                      JMP     $B6                     RETURN
001881                           ********************************************************************
001882                           * FORWARD SPACE A FILE
001883                           *
001884  05F7  D380 0FA2          FF      LNJ     $B5,<LABEL
001885  05F9  4646                       DC      'FF'                    LABEL1
001886  05FA  D380 13B3                  LNJ     $B5,<FSF                START IO FORWARD SPACE A FILE
001887  05FC  D380 0FEC                  LNJ     $B5,<INSTAT             WAIT FOR RDY, GET STAT
001888  05FE  9800 21B4                  LDR     $R1,<STAT1
001889  0600  9970 8400                  CMR     $R1,=Z'8400'            RDY, FM DET
001890  0602  0988                       BNE     >FFFA                   B IF STAT1 NG
001891  0603  9800 21B5                  LDR     $R1,<STAT2
001892  0605  9630 20FA                  XOR     $R1,<DXDENS,$R3         MASK WITH DENSITY BIT
001893  0607  9970 8000                  CMR     $R1,=Z'8000'            ON LINE
001894  0609  0904                       BE      >FFB                    B IF STAT OK
001895  060A  D380 1030          FFFA    LNJ     $B5,<FNER               STAT NG AFTER FSF
001896  060C  3331               FNER31  DC      '31'                    LABEL2
001897                           *
001898  060D  D380 100F          FFB     LNJ     $B5,<SENSE              NEXT?
001899  060F  8386                      JMP     $B6                     RETURN
001900                           ********************************************************************
001901                           * ERASE THE FIRST FILE MARK, BACKSPACE A REC. SHOULD GET TO BOT.
001902                           *
001903  0610  D380 0FA2          EF      LNJ     $B5,<LABEL
001904  0612  4546                       DC      'EF'                    LABEL1
001905  0613  8180 17E6          EFJ     IOLD    <WBOF,<IOWRIT,<DC32     (WRITE SOME DATA)
      0615  0000 22BC
      0617  0000 20F0
001906  0619  07FA                       BIOF    >EFJ
001907  061A  8000 22A2          EFB     IO      <WRITE,<OTTASK
      061C  0000 22C2
001908  061E  07FC                       BIOF    >EFB
001909  061F  D380 136E                  LNJ     $B5,<GETRNG             WAIT FOR N/BSY
001910                           *
001911  0621  D380 142B                  LNJ     $B5,<BSRW               BACK SPACE THAT RECORD
001912  0623  8280 21B4                  LD      <STAT1,=Z'0400'         FM
      0625  0400
001913  0626  0586                       BBF     >EFA                    B IF N/FM, OK
001914  0627  D380 1030                  LNJ     $B5,<FNER               SHOULDN'T BE FM ON NORM REC
001915  0629  3236               FNER26  DC      '26'                    LABEL2
001916  062A  0F80 0684                  B       <EF1                    ABORT THE TEST
001917                           *
001918                           * NOW ERASE THE CURRENT RECORD. DOING A BSRW SHOULD THEN FIND A FM.
001919                           *
001920  062C  D380 134D          EFA     LNJ     $B5,<ERAW               ERASE AND CHECK STAT
001921  062E  D380 142B                  LNJ     $B5,<BSRW               BACKSPACE, SHOULD GET TO BOT
001922  0630  9800 21B4                  LDR     $R1,<STAT1
001923  0632  9570 0600                  AND     $R1,=Z'0600'            FM, BOT
001924  0634  9970 0400                  CMR     $R1,=Z'0400'            SHOULD BE FM, N/BOT
001925  0636  0904                       BE      >EFC                    B IF STAT OK
001926  0637  D380 1030                  LNJ     $B5,<FNER               STAT1 SHOULD BE FM, N/BOT
001927  0639  3332               FNER32  DC      '32'                    LABEL2
001928                           *
001929                           * NOW GO TO BOT, DO BSR AND BSF. SHOULD DO NORMAL TERMINATION FOR BOTH.
001930                           *
001931  063A  D380 174B          EFC     LNJ     $B5,<GOBOT              REWIND
001932  063C  8000 229C                  IO      <REVREC,<OTTASK         START IO BACKSPACE
      063E  0000 22C2
001933  0640  07FA                       BIOF    >EFC
001934  0641  D380 0FC5                  LNJ     $B5,<TIME               WAIT FOR...
001935  0643  0001                       DC      1                       1 MILLISECOND
001936  0644  8000 21B4                  IO      <STAT1,<INSTW1          TRY TO GET STATUS
      0646  0000 22CC
001937  0648  0705                       BIOT    >EFD                    B IF ACK'ED (OK)
001938  0649  D380 1030                  LNJ     $B5,<FNER               SHOULDN'T TAKE TIME TO BSR AT BOT
001939  064B  3333               FNER33  DC      '33'                    LABEL2, STAT WORDS IN MSG ARE INVALID
001940  064C  0F94                       B       >EFF                    BYPASS THE NEXT TEST
001941                           *
001942  064D  8000 21B5          EFD     IO      <STAT2,<INSTW2
      064F  0000 22CD
001943  0651  9800 21B4                  LDR     $R1,<STAT1
```

```
001944  0653  9970 8200              CMR   $R1,=Z'8200'        RDY, BOT
001945  0655  0988                   BNE   >EFE               B IF STW1 NG
001946  0656  9800 21B5              LDR   $R1,<STAT2
001947  0658  9630 20FA              XOR   $R1,<DXDENS.$R3    MASK WITH DENSITY BIT
001948  065A  9970 8000              CMR   $R1,=Z'8000'       ON LINE
001949  065C  0904                   BE    >EFF               B IF STAT OK
001950  065D  D380 1030        EFE    LNJ   $B5,<FNER          STAT1 NG AFTER BSR FROM BOT
001951  065F  3334             FNER34 DC    '34'               LABEL2
001952                                *
001953                                * REPEAT PREVIOUS TEST FOR BSF.
001954                                *
001955  0660  8000 229E        EFF    IO    <REVFIL,<OTTASK    START IO BACKSPACE
        0662  0000 22C2
001956  0664  07FC                         BIOF  >EFF
001957  0665  D380 0FC5                   LNJ   $B5,<TIME          TIMEOUT FOR...
001958  0667  0001                   DC    1                  1 MILLISECOND
001959  0668  8000 21B4              IO    <STAT1,<INSTW1     TRY TO GET STATUS
        066A  0000 22CC
001960  066C  0705                         BIOT  >EFG               B IF ACK'ED (OK)
001961  066D  D380 1030              LNJ   $B5,<FNER          SHOULDN'T TAKE TIME TO BSF AT BOT...
001962  066F  3335             FNER35 DC    '35'               LABEL2, STAT WORDS IN MSG ARE INVALID
001963  0670  0F94                   B     >EF1               BYPASS NEXT TEST
001964                                *
001965  0671  8000 21B5        EFG    IO    <STAT2,<INSTW2
        0673  0000 22CD
001966  0675  9800 21B4              LDR   $R1,<STAT1
001967  0677  9970 8200              CMR   $R1,=Z'8200'       RDY, BOT
001968  0679  0988                   BNE   >EFM               B IF STAT1 NG
001969  067A  9800 21B5              LDR   $R1,<STAT2
001970  067C  9630 20FA              XOR   $R1,<DXDENS.$R3    MASK WITH DENSITY BIT
001971  067E  9970 8000              CMR   $R1,=Z'8000'       ON LINE
001972  0680  0904                   BE    >EF1               B IF STAT OK
001973  0681  D380 1030        EFM    LNJ   $B5,<FNER          STAT NG AFTER BSF FROM BOT
001974  0683  3434             FNER44 DC    '44'               LABEL2
001975                                *
001976  0684  D380 100F        EF1    LNJ   $B5,<SENSE         NEXT?
001977  0686  8386                   JMP   $B6                RETURN
001978                                ********************************************************************
001979                                * INTERRUPT TEST. CHECK ALL LEVELS OF CP RUPT AGAINST ALL LEVELS
001980                                * OF PERIPHERAL RUPT CONTROL. R5=DVC LVL, R6=CP LVL (OUTER LOOP).
001981                                *
001982  0687  EF60 0749              RP    STB   $B6,<RPB6          SAVE RETURN ADDRESS
001983  0689  8F00 0748                   STR   $R3,<RPR3          SAVE INDEX FOR FUTURE USE
001984  068B  D360 0FA2                   LNJ   $B5,<LABEL
001985  068D  5250                   DC    'RP'               LABEL1
001986  068E  0F7F                   NOP   >$-1               FOR DEBUG ONLY ???
001987  068F  0F7F                   NOP   >$-1               FOR DEBUG ONLY ???
001988  0690  8C51                   STS   =$R1
001989  0691  9A80 14E5              SRM   $R1,<CPCHAN,=Z'03C0'
        0693  03C0
001990  0694  AB80 0000        X     LAB   $B2,<ZHISAZ
001991  0696  BB80 1501              LAB   $B3,<ISAKDV        SAVE AREA FOR ALL LEVELS
001992  0698  BFF2             RPB   STB   $B3,+$B2           SET UP ALL LEVELS
001993  0699  AD80 14E9              CMB   $B2,<HOOBF         DONE YET?
001994  069B  0AF0                   BALE  >RPB               NO - KEEP GOING
001995  069C  9B80 1516              LAB   $B1,<DEVIH         ADDRESS OF RUPT HANDLER
001996  069E  9F80 1504              STB   $B1,<ISARP         PUT INTO SAVE AREA
001997  06A0  6C3F                   LDV   $R6,=63            STARTING LEVEL FOR PROGRAM
001998                                *
001999                                *
002000                                *
002001  06A1  8700 0000        X     RPC   CL    <ZHIAFB              RESET ALL ACTIVITY FLAGS
002002  06A3  8700 0001        X           CL    <ZHIAFB+1
002003  06A5  8700 0002        X           CL    <ZHIAFB+2
002004  06A7  8700 0003        X           CL    <ZHIAFB+3
002005  06A9  9800 14E3              LDR   $R1,<LVWC          QUICK LEVEL CHANGE
002006  06AB  9456                   OR    $R1,=$R6           GET READY FOR LEVEL CHANGE
002007  06AC  8E51             LEVXC LEV   =$R1               MAKE THE CHANGE
002008  06AD  9570 003F              AND   $R1,=Z'003F'       LOOK AT LEVEL ONLY
002009  06AF  8AD1                   INC   =$R1               GO BACK TO OLD LEVEL
002010  06B0  8810 0000        X     LBF   <ZHIAFB.$R1        AND RESET ACTIVITY FLAG FOR THAT LVL
002011  06B2  9856                   LDR   $R1,=$R6           GET CURRENT LEVEL BACK
002012  06B3  9270 003F              SUB   $R1,=63            IS THIS FIRST TIME THRU?
002013  06B5  1986                   BNEZ  $R1,>RPD           - BRANCH IF NOT FIRST TIME
002014  06B6  9B80 14EB              LAB   $B1,<ISATDV
002015  06B8  9F80 003F        X     STB   $B1,<ZHISAZ+63*$AF SETUP SAVE AREA AFTER LEV INSTR
002016  06BA  0F87                   B     >RPE               SKIP THIS CODE NOW
002017  06BB  9856             RPD   LDR   $R1,=$R6           RESTORE CURRENT LEVEL
002018  06BC  8AD1                   INC   =$R1               LOOK AT OLD LEVEL
002019  06BD  9B80 1501              LAB   $B1,<ISAKDV
002020  06BF  9F90 0000        X     STB   $B1,<ZHISAZ.$R1
002021  06C1  D870 003E             RPE   LDR   $R5,=62            INITIAL LEVEL FOR DEVICE
002022                                *
002023                                *
002024                                *
002025  06C3  8700 14E4             RPF   CL    <DEVSEM              RESET SEMAPHORE
002026  06C5  5901 006C              BEZ   $R5,RPN            B IF END OF TEST
002027  06C7  D400 14E5              OR    $R5,<CPCHAN        STICK IN CP CHANNEL NUMBER
002028  06C9  8055                   IO    =$R5,<OTRUPT       SET NEW DEVICE LEVEL
        06CA  0000 22BF
002029  06CC  07FD                         BIOF  >RPFA
002030  06CD  D570 003F              AND   $R5,=Z'003F'       LOOK AT DEVICE LEVEL
002031  06CF  DF00 14E6              STR   $R5,<DLVL          STORE IT AWAY
002032  06D1  EF00 14E8              STR   $R6,<PLVL          STORE PROGRAM LEVEL
002033  06D3  EF00 14E7              STR   $R6,<CLVL          **********
002034  06D5  F870 0080              LDR   $R7,=128           SETUP FOR TIME DELAY
002035  06D7  8000 22B2              IO    <STOPIO,<OTCONT    GENERATE AN INTERRUPT
        06D9  0000 22BE
002036  06DB  7701 FFFF             BDEC  $R7,$              WAIT HERE FOR A WHILE FOR RUPT
002037  06DD  0F01 FFFF              NOP   $                  MAY COME HERE AFTER RUPT
002038  06DF  D570 003F              AND   $R5,=Z'003F'       LOOK AT DEVICE LEVEL ONLY
002039  06E1  9855                   LDR   $R1,=$R5           PUT DEVICE LEVEL INTO R1
002040  06E2  1901 004F              BEZ   $R1,RPN            LEVEL ZERO CANT RUPT
002041  06E4  9256                   SUB   $R1,=$R6           SEE WHICH LEVEL WAS HIGHEST
002042  06E5  188A                   BGEZ  $R1,>RPJ           DEVICE SHOULDN'T HAVE RUPTED - BRANCH
002043  06E6  8980 14E4              CMZ   <DEVSEM            DID A RUPT HAPPEN?
002044  06E8  0981 0049              BNE   RPN                YES - EVERYTHINGS OK **********
002045  06EA  B800 0748              LDR   $R3,<RPR3
002046  06EC  D380 1030              LNJ   $B5,<FNER          SHOULD HAVE RUPTED BUT DIDN'T
002047  06EE  3336                   DC    '36'               LABEL2
002048  06EF  8980 14E4             RPJ   CMZ   <DEVSEM            HAS RUPT HAPPENED?
002049  06F1  0906                   BE    >RPK               NO - IT'S OK
002050  06F2  B800 0748              LDR   $R3,<RPR3
```

```
002051  06F4  D380 1030            LNJ   $B5,<FNER          RUPTED WHEN IT SHOULDN'T HAVE
002052  06F6  3337                FNER37 DC    '37'          LABEL2
002053  06F7  5901 003A            KPK   BEZ   $R5,KPN        SKIP THIS LEVEL ZERO CAN'T RUPT
002054  06F9  D970 003F                  CMR   $R5,=63         IS DEVICE AT LEVEL 63?
002055  06FB  0901 0036                  BE    KPN             YES - SKIP SINCE 63 CAN'T RUPT
002056  06FD  EF00 14E7                  STR   $R6,<CLVL       CURRENT PROGRAM LEVEL
002057  06FF  9870 003F                  LDR   $R1,=63         SET FUTURE LEVEL
002058  0701  9F00 14E8                  STR   $R1,<PLVL       STORE IN PROGRAM LEVEL
002059  0703  9470 8000                  OR    $R1,=Z'8000'    ACTION FOR LEV INSTR
002060  0705  9B80 070A                  LAB   $B1,<KPK1       CHANGE ADDRESS IN ISA
002061  0707  9F80 1504                  STB   $B1,<ISAKP      STORE IN P COUNTER
002062  0709  8E51                LEVXD  LEV   =$R1            DROP TO 63 TO ENABLE PENDING RUPT
002063  070A  0F01 FFFF            KPK1   NOP   $              RETURN TO HERE
002064  070C  F870 0080                  LDR   $R7,=128        SET UP FOR DELAY
002065  070E  7701 FFFF                  BDEC  $R7,$           WAIT HERE FOR WHILE
002066  0710  D800 14E6                  LDR   $R5,<DLVL       RESTORE DEVICE LEVEL TO R5
002067  0712  8800 0003      X           LBF   <ZH1AFB+3,=Z'0001'  SHUT OFF LEVEL 63 BIT
        0714  0001
002068  0715  8C52                      STS   =$R2
002069  0716  A570 003F                  AND   $R2,=Z'003F'
002070  0718  A955                      CMR   $R2,=$R5
002071  0719  0906                      BE    >KPK2
002072  071A  D800 0748                  LDR   $R3,<KPK3
002073  071C  D380 1030                  LNJ   $B5,<FNER       DIDN'T RUPT WHEN CP LVL WENT TO 63
002074  071E  3338                FNER38 DC    '38'          LABEL2
002075  071F  D900 14E7            KPK2   CMR   $R5,<CLVL       ARE THE PROG AND DEV AT SAME LVL?
002076  0721  0906                      BE    >KPK3           YES - NO NEED FOR LEV
002077  0722  C670 8080                  LDR   $R4,=Z'8080'
002078  0724  C400 14E7                  OR    $R4,<CLVL
002079  0726  8E54                LEVXE  LEV   =$R4
002080  0727  9B80 1516            KPK3   LAB   $B1,<DEVIH       RESTORE POINTER TO HANDLER
002081  0729  9F80 1504                  STB   $B1,<ISAKP
002082  072B  E800 14E7                  LDR   $R6,<CLVL       RESTORE PROG LEVEL TO R6
002083  072D  9856                      LDR   $R1,=$R6        PUT INTO R1
002084  072E  9B80 14EB                  LAB   $B1,<ISATDV     ISA FOR PROGRAM
002085  0730  9F90 0000      X           STB   $B1,<ZH1SAZ.$R1 RESTORE INTERRUPT VECTOR
002086  0732  5700 06C3            KPN    BDEC  $R5,<KPF        CHANGE DEVICE LEVEL AND BRANCH
002087  0734  6700 06A1                   BDEC  $R6,<KPC        CHANGE PROGRAM LEVEL AND BRANCH
002088                             *
002089  0736  B800 0748            KPM    LDR   $R3,<KPK3       RESTORE R3 FOR RETURN
002090  0738  D380 16EB                   LNJ   $B5,<INIZ       INITIALIZE
002091  073A  9B80 0745                   LAB   $B1,<KP0        CONTINUE AT KP0 AFTER "LEV"
002092  073C  9F80 230E                   STB   $B1,<SA15P
002093  073E  9B60 230C                   LAB   $B1,<SA15DV     INTERRUPT SAVE AREA
002094  0740  9F80 000F      X            STB   $B1,<ZH1SAZ+15*$AF  INTERRUPT VECTOR
002095  0742  8E70 800F            LEVXJ  LEV   =Z'8000'+15     SUSPEND TO CPU LVL 15
002096  0744  0000                      HLT                   "LEV" DIDN'T RUPT TO LVL 15
002097                             *
002098  0745  EC80 0749            KP0    LDB   $B6,<KPB6       RESTORE RETURN ADDRESS
002099  0747  8386                      JMP   $B6             RETURN
002100                             *
002101  0748  0000                KPK3   RESV  1,0             RESTORE R3 FOR INDEX
002102  0749  0000                KPB6   RESV  $AF,0
002103                             ********************************************************************
002104                             * STATUS TEST. CHECK ALL BITS OF BOTH STATUS WORDS.
002105                             * FIRST SEE THAT ALL BITS CAN BE SET AND THAT ALL BITS ARE RESET
002106                             * BY THEIR SPECIFIED INSTRUCTIONS.
002107                             *
002108  074A  D380 0FA2            SA     LNJ   $B5,<LABEL
002109  074C  5341                       DC    'SA'            LABEL1
002110  074D  D380 134D                   LNJ   $B5,<ERAW       GET OFF BOT
002111  074F  8070 FFFF            SAA    IO    =Z'FFFF',<OTSIW1  FORCE STATUS WORD 1 TO FFFF
        0751  0000 22C0
002112  0753  07FC                       BIOF  >SAA
002113  0754  8070 FFFF            SAB    IO    =Z'FFFF',<OTSIW2  FORCE STATUS WORD 2 TO FFFF
        0756  0000 22C1
002114  0758  07FC                       BIOF  >SAB
002115                             *
002116  0759  8700 21B4                   CL    <STAT1
002117  075B  8700 21B5                   CL    <STAT2
002118  075D  D380 0FC5                   LNJ   $B5,<TIME       TIMEOUT TO ALLOW FOR POLLING
002119  075F  0002                       DC    2               2 MSECS
002120  0760  D380 0FEC                   LNJ   $B5,<INSTAT     GET STATUS
002121  0762  A800 21B4                   LDR   $R2,<STAT1
002122  0764  A970 FCFF                   CMR   $R2,=Z'FCFF'    N/BOT, N/EOT
002123  0766  0988                       BNE   >SAC            B IF STAT1 NG
002124  0767  A800 21B5                   LDR   $R2,<STAT2
002125  0769  A630 20FA                   XOR   $R2,<DXDENS.$R3  MASK WITH DENSITY BIT
002126  076B  A970 8FFF                   CMR   $R2,=Z'8FFF'    N/RWND, N/PROTECT, N/HI DENS
002127  076D  0904                       BE    >SAD            B IF STAT OK
002128  076E  D380 1030            SAC    LNJ   $B5,<FNER       STAT SB FCFF-8FFF
002129  0770  3435                FNER45 DC    '45'            LABEL2
002130                             *
002131  0771  D380 0FEC            SAD    LNJ   $B5,<INSTAT     GET STAT. SHOULD RESET SOME BITS
002132  0773  9800 21B4                   LDR   $R1,<STAT1
002133  0775  9970 BCF9                   CMR   $R1,=Z'BCF9'    DID WE RESET ATTN, N/EXIST RES, BUS PAR
002134  0777  0988                       BNE   >SAE            B IF STAT1 NG
002135  0778  9800 21B5                   LDR   $R1,<STAT2
002136  077A  9630 20FA                   XOR   $R1,<DXDENS.$R3  MASK WITH DENSITY BIT
002137  077C  9970 8FFF                   CMR   $R1,=Z'8FFF'
002138  077E  0904                       BE    >SAF            B IF STAT OK
002139  077F  D380 1030            SAE    LNJ   $B5,<FNER       STAT SB BCF9-8FFF
002140  0781  3436                FNER46 DC    '46'            LABEL2
002141                             *
002142  0782  8070 FFFF            SAF    IO    =Z'FFFF',<OTSIW1  SET STAT1 AGAIN
        0784  0000 22C0
002143  0786  07FC                       BIOF  >SAF
002144  0787  D380 0FC5                   LNJ   $B5,<TIME       TIME TO POLL, BUT WON'T BECAUSE THERE'S...
002145  0789  0002                       DC    2               ...NO CHANGE IN "RWND" STAT BIT
002146  078A  8000 22A5                   IO    <NOPER,<OTTASK  TASK WORD SHOULD NOT BE ACCEPTED
        078C  0000 22C2
002147  078E  D380 0FEC                   LNJ   $B5,<INSTAT     GET STAT
002148  0790  9800 21B4                   LDR   $R1,<STAT1      SHOULD BE BOT, EOT BECAUSE DIDN'T POLL
002149  0792  9970 FFFF                   CMR   $R1,=Z'FFFF'    BUS PAR. ERR SHOULD PREVENT OTTASK
002150  0794  0988                       BNE   >SAH            B IF NG
002151  0795  9800 21B5                   LDR   $R1,<STAT2
002152  0797  9630 20FA                   XOR   $R1,<DXDENS.$R3  MASK WITH DENSITY BIT
002153  0799  9970 8FFF                   CMR   $R1,=Z'8FFF'    STILL 8FFF FROM TEST "SAD"
002154  079B  0904                       BE    >SAG            B IF STAT OK
002155  079C  D380 1030            SAH    LNJ   $B5,<FNER       STAT SB FFFF-8FFF
002156  079E  3437                FNER47 DC    '47'            LABEL2
002157                             *
002158  079F  8070 FFFD            SAG    IO    =Z'FFFD',<OTSIW1  SET STAT1 =1'S EXCEPT "BUS PAR"
```

```
                07A1  0000 22C0
002159          07A3  07FC                    BIOF    >SAG
002160          07A4  8070 FFFF       SAI     IO      =Z'FFFF',<OTSTW2        SET STATUS BITS
                07A6  0000 22C1
002161          07A8  07FC                    BIOF    >SAI
002162          07A9  D380 0FC5               LNJ     $B5,<TIME               TIMEOUT FOR POLLING
002163          07AB  0002                    DC      2                       2 MSEC
002164          07AC  8000 22A5               IO      <NOPER,<OTTASK
                07AE  0000 22C2
002165          07B0  D380 0FEC               LNJ     $B5,<INSTAT
002166          07B2  9800 21B4               LDR     $R1,<STAT1
002167          07B4  8980 20E9               CMZ     <BDCFLG
002168          07B6  0908                    BE      >SAK                    B IF BDC2
002169          07B7  A800 2180               LDR     $R2,<FIRM               GET FIRMWARE REV NMBR
002170          07B9  2021                    CMV     $R2,=X'21'              REV '21
002171          07BA  0204                    BL      >SAK                    B IF BDC3 AND BEFORE REV '21
002172          07BB  9970 C000               CMR     $R1,=Z'C000'            ATTN BIT
002173          07BD  0F83                    B       >SAL                    CONTINUE
002174          07BE  9970 8000       SAK     CMR     $R1,=Z'8000'
002175          07C0  0988            SAL     BNE     >SAJ                    B IF NG
002176          07C1  9800 21B5               LDR     $R1,<STAT2
002177          07C3  9630 20FA               XOR     $R1,<DXDENS,$R3         MASK WITH DENSITY BIT
002178          07C5  9970 8000               CMR     $R1,=Z'8000'
002179          07C7  0904                    BE      >SB                     B IF OK
002180          07C8  D380 1030       SAJ     LNJ     $B5,<FNER               STAT SB 8000-8000 AFTER OTTASK
002181          07CA  3635            FNER65  DC      '65'                    LABEL2
002182                                *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002183                                * BIT RESETTING HAS BEEN CHECKED. NOW CHECK OPERATION OF INDIVIDUAL
002184                                * BITS. FIRST READ REVERSE. CHECK FOR APPROPRIATE ERRORS (OP CHECK,
002185                                * FUNCTIONALITY NOT AVAILABLE).
002186                                *
002187          07CB  D380 0FA2       SB      LNJ     $B5,<LABEL
002188          07CD  5342                    DC      'SB'                    LABEL1
002189          07CE  D380 174B               LNJ     $B5,<GOBOT              REWIND
002190          07D0  D380 134D               LNJ     $B5,<ERAW               GET OFF BOT
002191          07D2  8000 22A3               IO      <READRV,<OTTASK         TRY TO READ BACKWARDS, SHOULDN'T.
                07D4  0000 22C2
002192          07D6  0F7F                    NOP     >$-1                    DELAY
002193          07D7  D380 0FEC               LNJ     $B5,<INSTAT
002194          07D9  9800 21B4               LDR     $R1,<STAT1
002195          07DB  9970 8010               CMR     $R1,=Z'8010'            RDY, OP CHECK
002196          07DD  0981 0008               BNE     SBA                     B IF NG
002197          07DF  9800 21B5               LDR     $R1,<STAT2
002198          07E1  9630 20FA               XOR     $R1,<DXDENS,$R3         MASK WITH DENSITY BIT
002199          07E3  9970 8010               CMR     $R1,=Z'8010'            ON LINE, FUNC N/AVAILABLE
002200          07E5  0904                    BE      >SBB                    B IF STAT OK
002201          07E6  D3C0 0849       SBA     LNJ     $B5,FNER                STAT SB 8010-8010 AFTER READ REVERSE
002202          07E8  3237            FNER27  DC      '27'                    LABEL2
002203                                *
002204                                * TRY TO WRITE REC LESS THAN MIN. ALLOWED LENGTH (FROM BOT).
002205                                * SHOULD SET OP CHECK AND NOT MOVE OFF BOT.
002206                                *
002207          07E9  D380 174B       SBB     LNJ     $B5,<GOBOT              REWIND
002208          07EB  8751                    CL      =$R1                    RANGE = 0
002209          07EC  8180 17E6               IOLD    <WBUF,<IOWRIT,=$R1
                07EE  0000 22BC
                07F0  0051
002210          07F1  07F8                    BIOF    >SBB
002211          07F2  8000 22A2       SBC     IO      <WRITE,<OTTASK          TRY TO WRITE
                07F4  0000 22C2
002212          07F6  07FC                    BIOF    >SBC
002213                                *
002214          07F7  D380 0FEC               LNJ     $B5,<INSTAT
002215          07F9  9800 21B4               LDR     $R1,<STAT1
002216          07FB  9970 8210               CMR     $R1,=Z'8210'            RDY, BOT, OP CHECK
002217          07FD  0988                    BNE     >SBD                    B IF NG
002218          07FE  9800 21B5               LDR     $R1,<STAT2
002219          0800  9630 20FA               XOR     $R1,<DXDENS,$R3         MASK WITH DENSITY BIT
002220          0802  9970 8000               CMR     $R1,=Z'8000'            ON LINE
002221          0804  0904                    BE      >SC                     B IF OK
002222          0805  D380 1030       SBD     LNJ     $B5,<FNER               STAT SB 8210-8000 FOR WRITE W/RNG=0
002223          0807  3238            FNER28  DC      '28'                    LABEL2
002224                                *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002225                                * WRITE DATA FROM BOT, THEN TRY TO READ THAT REC (WITH RANGE = 0).
002226                                * SHOULD GIVE SAME RESULTS AS DOING 'FORWARD SPACE A RECORD'.
002227                                * NOTE: IF CAN'T WRITE FIRST RECORD, ABORT TESTS SC AND SD.
002228                                * IF CAN'T READ FIRST RECORD, GO ON AND TRY REST OF TEST
002229                                * ANYWAY.
002230                                *
002231          0808  D380 0FA2       SC      LNJ     $B5,<LABEL
002232          080A  5343                    DC      'SC'                    LABEL1
002233          080B  D380 174B               LNJ     $B5,<GOBOT              REWIND
002234          080D  8730 20FE               CL      <DXFN,$R3               CLEAR FILE NMBR
002235          080F  8730 210E               CL      <DXRN,$R3               CLEAR REC NMBR
002236                                *
002237          0811  D380 10DE               LNJ     $B5,<WREC               WRITE A REC
002238          0813  000C                    DC      X'0C'                   DATA
002239          0814  0040                    DC      64                      RANGE (DEC)
002240          0815  17E6                    DC      <WBUF                   BUFFER
002241          0816  10C2                    DC      <RT4                    RETRY ROUTINE (NOP)
002242          0817  08AE                    DC      <SE                     RETRY LOC (ABORT)
002243          0818  10DD                    DC      <ABR3                   ABORT ROUTINE (NOP)
002244          0819  08AE                    DC      <SE                     ABORT LOCATION
002245                                *
002246          081A  D380 174B               LNJ     $B5,<GOBOT              REWIND
002247                                *
002248          081C  D380 10B6               LNJ     $B5,<SET                SET UP FOR READ
002249          081E  D380 118D               LNJ     $B5,<RREC               READ A RECORD
002250          0820  0040                    DC      64                      RANGE
002251          0821  1C66                    DC      <RBUF                   BUFFER
002252          0822  10C2                    DC      <RT4                    RETRY ROUTINE (NOP)
002253          0823  0826                    DC      <SCF                    RETRY LOCATION
002254          0824  10DD                    DC      <ABR3                   ABORT ROUTINE (NOP)
002255          0825  08AE                    DC      <SE                     ABORT LOCATION
002256                                *
002257          0826  D380 174B       SCF     LNJ     $B5,<GOBOT              REWIND
002258          0828  D380 175C               LNJ     $B5,<FRB33              FILL READ BUF W/'33'
002259          082A  8751                    CL      =$R1                    RANGE = 0
002260          082B  8180 1C66       SCA     IOLD    <RBUF,<IOREAD,=$R1
                082D  0000 22BB
                082F  0051
002261          0830  07FB                    BIOF    >SCA
002262          0831  8000 229F       SCB     IO      <READFD,<OTTASK
```

```
0833 0000 22C2
0835 07FC               BIOF   >SCB
0836 D380 136E          LNJ    $B5,<GETRNG          WAIT FOR RDY, GET RANGE
0838 D380 0FEC          LNJ    $B5,<INSTAT
083A 9800 21B4          LDR    $R1,<STAT1
083C 9970 8080          CMR    $R1,=Z'8080'         RDY, UNEQUAL LENGTH CHECK
083E 0988               BNE    >SCC                 B IF NG
083F 9800 21B5          LDR    $R1,<STAT2
0841 9630 20FA          XOR    $R1,<DXDENS.$R3      MASK WITH DENSITY BIT
0843 9970 8000          CMR    $R1,=Z'8000'
0845 0904               BE     >SCD                 B IF OK
0846 D380 1030   SCC    LNJ    $B5,<FNER            STAT SB 8080-8000 FOR READ W/RANGE=0
0848 3438        FNER48 DC     '48'                 LABEL2
                 *
0849 9800 1C66   SCD    LDR    $R1,<RBUF
084B 9900 21BF          CMR    $R1,<X3333           CHECK DATA (FIRST BYTE)
084D 0984               BNE    >SCE                 B IF DATA GOT TRANSFERED
084E 8980 219D          CMZ    <RANGE
0850 0904               BE     >SD                  B IF RANGE OK
0851 D380 1030   SCE    LNJ    $B5,<FNER            RNG SB =0, DATA SB ='3333'
0853 3439        FNER49 DC     '49'                 LABEL2
                 *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                 * UNEQUAL LENGTH CHECK. WORK WITH 64 BYTE RECORD PREVIOUSLY WRITTEN FROM
                 * BOT.  FIRST CHECK WITH TOO LARGE A RANGE. (NORMAL OPERATION ALREADY
                 * VERIFIED IN SUBROUTINE 'ONES' WHEN WRITTEN.)
                 *
0854 D380 0FA2   SD     LNJ    $B5,<LABEL
0856 5344               DC     'SD'                 LABEL1
0857 D380 174B          LNJ    $B5,<GOBOT           REWIND
0859 D380 175C          LNJ    $B5,<FRB33           FILL READ BUF W/'3333'
085B 9870 0041          LDR    $R1,=65              LONG RANGE (BYTES)
085D 8180 1C66   SDA    IOLD   <RBUF,<IOREAD,=$R1
085F 0000 22BB
0861 0051
0862 07FB               BIOF   >SDA
0863 8000 229F   SDB    IO     <READFD,<OTTASK
0865 0000 22C2
0867 07FC               BIOF   >SDB
0868 D380 136E          LNJ    $B5,<GETRNG          WAIT RDY, GET RANGE REMAINING
086A D380 0FEC          LNJ    $B5,<INSTAT
086C A800 219D          LDR    $R2,<RANGE
086E 2D01               CMV    $R2,=1
086F 0992               BNE    >SDC                 B IF 'RANGE' NOT EQUAL 1, NG
0870 88D1               DEC    =$R1
0871 A290 1C66          LLH    $R2,<RBUF.$R1        GET LAST BYTE OF DATA
0873 2D33               CMV    $R2,=X'33'
0874 098D               BNE    >SDC                 B IF LAST DATA TRANSFERED, NG
0875 9800 21B4          LDR    $R1,<STAT1
0877 9970 8080          CMR    $R1,=Z'8080'         RDY, UNEQUAL LENGTH CHECK
0879 0988               BNE    >SDC                 B IF   STAT1 NG
087A 9800 21B5          LDR    $R1,<STAT2
087C 9630 20FA          XOR    $R1,<DXDENS.$R3      MASK WITH DENSITY BIT
087E 9970 8000          CMR    $R1,=Z'8000'
0880 0904               BE     >SDD                 B IF OK
0881 D380 1030   SDC    LNJ    $B5,<FNER            RNG SB =1, STAT SB 8080-8000...
0883 3530        FNER50 DC     '50'                 LABEL2   ...AND LAST DATA BYTE SB =33
                 *
                 * UNEQUAL LENGTH CHECKS, USE TOO SHORT A RANGE
                 *
0884 D380 174B   SDD    LNJ    $B5,<GOBOT           REWIND
0886 D380 175C          LNJ    $B5,<FRB33           FILL BUF WITH 3333
0888 1C3F               LDV    $R1,=63              SHORT RANGE, BYTES
0889 8180 1C66   SDE    IOLD   <RBUF,<IOREAD,=$R1
088B 0000 22BB
088D 0051
088E 07FB               BIOF   >SDE
088F 8000 229F   SDF    IO     <READFD,<OTTASK
0891 0000 22C2
0893 07FC               BIOF   >SDF
0894 D380 136E          LNJ    $B5,<GETRNG          WAIT FOR RDY, GET RANGE
0896 D380 0FEC          LNJ    $B5,<INSTAT
0898 8980 219D          CMZ    <RANGE
089A 0991               BNE    >SDG                 B IF RANGE NOT EQUAL 0, NG
089B A290 1C66          LLH    $R2,<RBUF.$R1        GET LAST-PLUS-ONE DATA BYTE
089D 2D33               CMV    $R2,=X'33'
089E 098D               BNE    >SDG                 B IF WANGE + 1 WAS TRANSFERED, NG
089F 9800 21B4          LDR    $R1,<STAT1
08A1 9970 8080          CMR    $R1,=Z'8080'         RDY, UNEQUAL LENGTH CHECK
08A3 0988               BNE    >SDG                 B IF STAT1 NG
08A4 9800 21B5          LDR    $R1,<STAT2
08A6 9630 20FA          XOR    $R1,<DXDENS.$R3      MASK WITH DENSITY BIT
08A8 9970 8000          CMR    $R1,=Z'8000'         ON LINE
08AA 0904               BE     >SE                  B IF OK
08AB D380 1030   SDG    LNJ    $B5,<FNER            RNG SB=0, STAT SB 8080-8000...
08AD 3531        FNER51 DC     '51'                 LABEL2   ...LAST BYTE+1 SB=33
                 *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                 * VERIFY THAT AN 'OPERATION CHECK' RESULTS WHEN THE IOLD INDICATES A
                 * READ OPERATION AND THE TASK INDICATES A WRITE OPERATION. START AT
                 * BOT, AND SHOULDN'T MOVE OFF BOT.
                 *
08AE D380 0FA2   SE     LNJ    $B5,<LABEL
08B0 5345               DC     'SE'                 LABEL1
08B1 D380 174B          LNJ    $B5,<GOBOT           REWIND
08B3 1C40               LDV    $R1,=64              RANGE
08B4 8180 1C66          IOLD   <RBUF,<IOREAD,=$R1   (SET UP FOR READ OP)
08B6 0000 22BB
08B8 0051
08B9 07F5               BIOF   >SE
08BA 8000 22A2   SEA    IO     <WRITE,<OTTASK       ACTUALLY TELL IT TO WRITE (A CONFLICT)
08BC 0000 22C2
08BE 07FC               BIOF   >SEA
08BF D380 0FEC          LNJ    $B5,<INSTAT
08C1 9800 21B4          LDR    $R1,<STAT1
08C3 9970 8210          CMR    $R1,=Z'8210'         RDY, BOT, OP CHECK
08C5 0988               BNE    >SEB                 B IF STAT1 NG
08C6 9800 21B5          LDR    $R1,<STAT2
08C8 9630 20FA          XOR    $R1,<DXDENS.$R3      MASK WITH DENSITY BIT
08CA 9970 8000          CMR    $R1,=Z'8000'         ON LINE
08CC 0904               BE     >SF                  B IF STAT OK
08CD D380 1030   SEB    LNJ    $B5,<FNER            STAT SB 8210-8000 FOR WRITE/READ CONFLICT
08CF 3532        FNER52 DC     '52'                 LABEL2
                 *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                 * CHECK OPERATION OF "ANSI INHIBIT". FIRST WRITE A "SHORT" RECORD,
```

```
002366                                  * FOLLOWED BY A FILE MARK. (BYPASS IF P.E. DRIVE)
002367                                  *
002368  08D0  D380 0FA2                 SF      LNJ     $B5,<LABEL
002369  08D2  5346                              DC      'SF'                    LABEL1
002370  08D3  8280 20E9                         LB      <BDCFLG,=1
        08D5  0001
002371  08D6  0585                              BBF     >SFH                    B IF BDC2
002372  08D7  8980 20FA                         CMZ     <DXDENS,$R3             CHECK DENSITY FLAG
002373  08D9  0980 09A3                         BNE     <SH                     B IF THIS IS A P.E. DRIVE
002374  08DB  D380 174B                 SFH     LNJ     $B5,<GOBOT              REWIND
002375  08DD  8000 22A6                         IO      <ANSINH,<OTCONF         (INHIBIT ANSI)
        08DF  0000 22BD
002376  08E1  07EF                              BIOF    >SF
002377                                  *
002378  08E2  9870 AAAA                         LDR     $R1,=Z'AAAA'            DATA
002379  08E4  9F00 17E6                         STR     $R1,<WBUF
002380  08E6  8180 17E6                 SFA     IOLD    <WBUF,<IOWRIT,<DC2      (RANGE = 2)
        08E8  0000 22BC
        08EA  0000 20ED
002381  08EC  07FA                              BIOF    >SFA
002382  08ED  8000 22A2                 SFB     IO      <WRITE,<OTTASK          (WRITE SHORT REC, RNG = 2)
        08EF  0000 22C2
002383  08F1  07FC                              BIOF    >SFB
002384  08F2  D380 0FEC                         LNJ     $B5,<INSTAT
002385  08F4  9800 21B4                         LDR     $R1,<STAT1
002386  08F6  9970 8000                         CMR     $R1,=Z'8000'            RDY
002387  08F8  0988                              BNE     >SFC                    B IF STAT1 NG
002388  08F9  9800 21B5                         LDR     $R1,<STAT2
002389  08FB  9630 20FA                         XOR     $R1,<DXDENS,$R3         MASK WITH DENSITY BIT
002390  08FD  9970 8000                         CMR     $R1,=Z'8000'            ON LINE
002391  08FF  0904                              BE      >SFD                    B IF STAT OK
002392  0900  D380 1030                 SFC     LNJ     $B5,<FNER               STAT SB 8000-8000 AFTER ANSI INH WRITE
002393  0902  3533                      FNER53  DC      '53'                    LABEL2
002394                                  *
002395  0903  8000 22A9                 SFD     IO      <NORMAL,<OTCONF         RESET ANSI INH
        0905  0000 22BD
002396  0907  07FC                              BIOF    >SFD
002397  0908  D380 131A                         LNJ     $B5,<WFM                WRITE FILE MARK
002398  090A  D380 174B                         LNJ     $B5,<GOBOT              REWIND
002399                                  *
002400                                  * NOW FORWARD SPACE A REC IN NORMAL MODE. SHOULD BYPASS THE
002401                                  * SHORT "NOISE" RECORD AND GET TO THE FILE MARK.
002402                                  *
002403  090C  D380 13E9                         LNJ     $B5,<FSK                START IO FORWARD SPACE A REC...
002404  090E  D380 0FEC                         LNJ     $B5,<INSTAT             WAIT FOR RDY, GET STATUS
002405  0910  9800 21B4                         LDR     $R1,<STAT1
002406  0912  9970 8C00                         CMR     $R1,=Z'8C00'            RDY, COR MED ERR, FM DET, N/BOT
002407  0914  0981 0008                         BNE     SFE                     B IF STAT1 NG
002408  0916  9800 21B5                         LDR     $R1,<STAT2
002409  0918  9630 20FA                         XOR     $R1,<DXDENS,$R3         MASK WITH DENSITY BIT
002410  091A  9970 8000                         CMR     $R1,=Z'8000'            ON LINE
002411  091C  0904                              BE      >SFF                    B IF STAT OK
002412  091D  D380 1030                 SFE     LNJ     $B5,<FNER               STAT SB 8C00-8000 FOR NON-ANSI READ...
002413  091F  3534                      FNER54  DC      '54'                    LABEL2   ...OF SHORT REC, FM DETECTED
002414                                  *
002415                                  * REPEAT FORWARD SPACE FROM BOT, BUT WITH ANSI INHIBITED.
002416                                  * SHOULD STOP AFTER "SHORT" RECORD AND NOT GET TO FM DETECT.
002417                                  *
002418  0920  D380 174B                 SFF     LNJ     $B5,<GOBOT              REWIND
002419  0922  8000 22A6                         IO      <ANSINH,<OTCONF         (INHIBIT ANSI MODE)
        0924  0000 22BD
002420  0926  07FA                              BIOF    >SFF
002421  0927  D380 13E9                         LNJ     $B5,<FSK                START IO SPACE
002422  0929  D380 0FEC                         LNJ     $B5,<INSTAT             WAIT RDY, GET STAT
002423  092B  9800 21B4                         LDR     $R1,<STAT1
002424  092D  9970 8000                         CMR     $R1,=Z'8000'            RDY, N/FM DET
002425  092F  0988                              BNE     >SFG                    B IF STAT NG
002426  0930  9800 21B5                         LDR     $R1,<STAT2
002427  0932  9630 20FA                         XOR     $R1,<DXDENS,$R3         MASK WITH DENSITY BIT
002428  0934  9970 8000                         CMR     $R1,=Z'8000'            B IF OK
002429  0936  0904                              BE      >SG                     B IF OK
002430  0937  D380 1030                 SFG     LNJ     $B5,<FNER               STAT SB 8000-8000 FOR READ SHORT REC..
002431  0939  3535                      FNER55  DC      '55'                    LABEL2   ...WITH ANSI INHIBITED
002432                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002433                                  * LRC, CRC, VRC ERROR TEST (SKIP IF P.E. DRIVE)
002434                                  *
002435  093A  D380 0FA2                 SG      LNJ     $B5,<LABEL
002436  093C  5347                              DC      'SG'                    LABEL1
002437  093D  2C00                              LDV     $R2,=0                  CLEAR THE INDEX
002438  093E  9DF0 FFFF                         LDH     $R1,=-1                 DATA (=FF)
002439  0940  9B60 17E6                         LAB     $B1,<WBUF
002440  0942  9F80 21B3                         STB     $B1,<SLDB               SAVE FOR ERROR REPORT
002441                                  *
002442  0944  97ED                      SGA     STH     $R1,$B1.+$R2            FILL FIRST 31 BYTES WITH 'FF'
002443  0945  2D1F                              CMV     $R2,=31
002444  0946  027E                              BL      >SGA                    KEEP LOOPING
002445                                  *
002446  0947  87EB                      SGB     CLH     $B1.+$R2                CLEAR REST OF BUFFER
002447  0948  A970 0800                         CMR     $R2,=2048
002448  094A  027D                              BL      >SGB                    KEEP LOOPING
002449                                  *
002450  094B  97C1 0011                         STH     $R1,$B1.17              FILL IN PHONEY CRC (34TH BYTE)
002451  094D  97C1 0013                         STH     $R1,$B1.19              FILL FALSE LRC (38TH BYTE)
002452                                  *
002453                                  * BUFFER FILLED, NOW WRITE REC IN DIAGNOSTIC MODE
002454                                  *
002455  094F  D380 174B                         LNJ     $B5,<GOBOT              REWIND
002456  0951  8000 22AB                 SGC     IO      <EVNPAR,<OTCONF         (EVEN PAR, ANSI INH, DIAG MODE)
        0953  0000 22BD
002457  0955  07FC                              BIOF    >SGC
002458  0956  9800 20EE                         LDR     $R1,<DC4096             GET RANGE
002459  0958  9F00 219E                         STR     $R1,<RNG                SAVE FOR ERROR REPORT
002460  095A  8180 17E6                         IOLD    <WBUF,<IOWRIT,<DC4096
        095C  0000 22BC
        095E  0000 20EE
002461  0960  07F1                              BIOF    >SGC
002462  0961  8000 22A2                 SGD     IO      <WRITE,<OTTASK
        0963  0000 22C2
002463  0965  07FC                              BIOF    >SGD
002464                                  *
002465  0966  D380 136E                         LNJ     $B5,<GETRNG             WAIT FOR RDY
002466  0968  D380 0FEC                         LNJ     $B5,<INSTAT
002467  096A  9800 21B4                         LDR     $R1,<STAT1
```

```
002468  096C  9970 A040                    CMR   $R1,=Z'A040'          RDY, RETRY MED ERR, NON RETRY ERR.
002469  096E  0988                         BNE   >SGE                  B IF STAT NG
002470  096F  9800 21B5                    LDR   $R1,<STAT2
002471  0971  9630 20FA                    XOR   $R1,<DXDENS,$R3       MASK WITH DENSITY BIT
002472  0973  9970 8602                    CMR   $R1,=Z'8602'          ON LINE, VRC, CRC, EOB EARLY ERRORS
002473  0975  0904                         BE    >SGF                  B IF OK
002474  0976  D380 1030              SGE   LNJ   $B5,<FNER             STAT SB A040-8602 AFTER DIAG WRITE...
002475  0978  3536                   FNER56 DC   '56'                  LABEL2   ...WITH ARTIFICIAL CRC & LRC GAP
002476                               *
002477                               * REWIND AND READ IN NORMAL MODE. SHOULD HAVE VRC, CRC, AND LRC ERRORS.
002478                               *
002479  0979  D380 174B              SGF   LNJ   $B5,<GOBOT            REWIND
002480  097B  8000 22A9                    IO    <NORMAL,<OTCONF
      097D  0000 22BD
002481  097F  07FA                         BIOF  >SGF
002482  0980  1C1F                         LDV   $R1,=31               RANGE = 31
002483  0981  9F00 219E                    STR   $R1,<RNG              SAVE FOR ERROR REPORT
002484  0983  9B80 1C66                    LAB   $B1,<RBUF
002485  0985  9F80 21B3                    STB   $B1,<SLDB             SAVE FOR ERROR REPORT
002486  0987  8180 1C66              SGG   IOLD  <RBUF,<IOREAD,=$R1
      0989  0000 22BB
      098B  0051
002487  098C  07FB                         BIOF  >SGG
002488  098D  8000 229F              SGH   IO    <READFD,<OTTASK
      098F  0000 22C2
002489  0991  07FC                         BIOF  >SGH
002490  0992  D380 136E                    LNJ   $B5,<GETRNG           WAIT FOR RDY, GET RANGE
002491  0994  D380 0FEC                    LNJ   $B5,<INSTAT
002492  0996  9800 21B4                    LDR   $R1,<STAT1
002493  0998  9970 A000                    CMR   $R1,=Z'A000'          RDY, RETRY MED ERR
002494  099A  0986                         BNE   >SGI                  B IF STAT1 NG
002495  099B  9800 21B5                    LDR   $R1,<STAT2
002496  099D  9970 8700                    CMR   $R1,=Z'8700'          ON LINE, VRC, CRC, LRC ERR
002497  099F  0904                         BE    >SH                   B IF OK
002498  09A0  D380 1030              SGI   LNJ   $B5,<FNER             STAT SB A000-8700 AFTER READ REC...
002499  09A2  3537                   FNER57 DC   '57'                  LABEL2   ...WHICH WAS DIAG WRITTEN.
002500                               *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002501                               * CHECK DATA SERVICE RATE ERROR (ONLY WITH "BDC2" FIRMWARE)
002502                               *
002503  09A3  D380 0FA2              SH    LNJ   $B5,<LABEL
002504  09A5  5348                         DC    'SH'                  LABEL1
002505  09A6  8280 20E9                    LB    <BDCFLG,=1
      09A8  0001
002506  09A9  0500 09E3                    BBT   <SI                   SKIP THIS TEST IF "BDC3" FIRMWARE
002507                               *
002508  09AB  8000 22B3                    IO    <TESTMD,<OTCONT       (SET TEST MODE)
      09AD  0000 22BE
002509  09AF  8030 22AC                    IO    <LCHNO,$R3,<OTCONT    (OUTPUT LOGICAL CHAN NMBR)
      09B1  0000 22BE
002510  09B3  8000 22AA                    IO    <INDEX,<OTCONT        (SET INDEX MODE)
      09B5  0000 22BE
002511                               *
002512                               * SET WRITE, WRITE DATA SERVICE, WRITE DATA ENABLE BITS.
002513                               * DON'T PUT DATA INTO WRITE FIFO.
002514                               *
002515  09B7  8000 22B5                    IO    <WRENB,<OTCONT        (SET CONTROL REG ONE, BITS 2, 6, 7)
      09B9  0000 22BE
002516                               *
002517  09BB  D380 0FC5                    LNJ   $B5,<TIME             DELAY FOR AT LEAST ONE FRAME CNT
002518  09BD  0001                         DC    =1                    1 MS
002519                               *
002520                               * TRANSFER STAT BYTE 2 TO SCRATCH-PAD, THEN INPUT STAT AND CHECK.
002521                               *
002522  09BE  8000 22B6                    IO    <XFER1,<OTCONT        (SET SCRATCH-PAD ADR TO STAT 2)
      09C0  0000 22BE
002523  09C2  8000 22B7                    IO    <XFER2,<OTCONT        (XFER STAT BYTE TO SCRATCH-PAD)
      09C4  0000 22BE
002524  09C6  8000 22B8                    IO    <WRINH,<OTCONT        (RESET WRITE ORDER)
      09C8  0000 22BE
002525  09CA  8000 22B9                    IO    <CLADAP,<OTCONT       (ADAPTER HARD CLEAR)
      09CC  0000 22BE
002526                               *
002527  09CE  8000 22B1                    IO    <SCRDY,<OTCONT        (SET CHAN RDY)
      09D0  0000 22BE
002528  09D2  8000 22B4                    IO    <WAITLP,<OTCONT       (BRANCH TO WAIT LOOP)
      09D4  0000 22BE
002529  09D6  8000 22B0                    IO    <NOTEST,<OTCONT       (GET OUT OF TEST MODE)
      09D8  0000 22BE
002530                               *
002531  09DA  D380 0FEC                    LNJ   $B5,<INSTAT           GET STATUS
002532  09DC  8280 21B5                    LB    <STAT2,=Z'0800'       SERVICE RATE ERROR ?
      09DE  0800
002533  09DF  0504                         BBT   >SI                   B IF ERROR DETECTED, OK
002534  09E0  D380 1030                    LNJ   $B5,<FNER             SERVICE RATE ERROR N/DETECTED
002535  09E2  3538                   FNER58 DC   '58'                  LABEL2
002536                               *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002537                               * CHECK NON-EXISTANT RESOURCE ERROR (FROM BOT)
002538                               *
002539  09E3  D380 0FA2              SI    LNJ   $B5,<LABEL
002540  09E5  5349                         DC    'SI'                  LABEL1
002541  09E6  D380 174B                    LNJ   $B5,<GOBOT            REWIND
002542                                     CALL  ZV$RD,ZV$HM           GET HMA IN B7
      09E8  FBF0 0001
      09EA  D380 0000              X
002543  09EC  9BC7 FFF9                    LAB   $B1,$B7,-7
002544  09EE  9F80 21B3                    STB   $B1,<SLDB             ROOM FOR ONLY 16 BYTES OF DATA
002545  09F0  8181                   SIA   IOLD  $B1,<IOWRIT,<DC32     (RANGE OF 32 BYTES)
      09F1  0000 22BC
      09F3  0000 20F0
002546  09F5  07FB                         BIOF  >SIA
002547  09F6  8000 22A2              SIB   IO    <WRITE,<OTTASK
      09F8  0000 22C2
002548  09FA  07FC                         BIOF  >SIB
002549  09FB  D380 0FEC                    LNJ   $B5,<INSTAT           WAIT FOR N/BSY, GET STAT
002550  09FD  9800 21B4                    LDR   $R1,<STAT1
002551  09FF  9970 8004                    CMR   $R1,=Z'8004'
002552  0A01  098A                         BNE   >SIC                  B IF STAT NG
002553  0A02  9800 21B5                    LDR   $R1,<STAT2
002554  0A04  8851                         LBF   =$R1,=Z'0800'         IGNORE SERVICE RATE ERROR, IF ANY
      0A05  0800
002555  0A06  9630 20FA                    XOR   $R1,<DXDENS,$R3       MASK WITH DENSITY BIT
002556  0A08  9970 8000                    CMR   $R1,=Z'8000'          ON-LINE
002557  0A0A  0904                         BE    >SID                  B IF STAT OK
```

```
002558  0A0B  D380 1030          SIC     LNJ     $B5,<FNER           STAT NG, SB FUNCTION N/AVAIL
002559  0A0D  3539               FNER59  DC      '59'                LABEL2
002560                           *
002561  0A0E  D380 16EB          SID     LNJ     $B5,<INIZ           DO A BDC INITIALIZE
002562  0A10  D380 100F                  LNJ     $B5,<SENSE          NEXT?
002563  0A12  8386                       JMP     $B6                 RETURN
002564                           ********************************************************************
```

```
002565                                  /*****************************************************************
002566                                  *
002567                                  * TESTS AA-AS, GENERAL WRITE/READ ROUTINES
002568                                  *
002569                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002570                                  *
002571  0A13  D380 0FA2                 FILE    LNJ     $B5,<LABEL
002572  0A15  4649                      FI      DC      'FI'                    LABEL1
002573  0A16  8070 0000                         IO      =0,<OTRUPT              INHIBIT INTERRUPTS
        0A18  0000 22BF
002574  0A1A  D380 174B                          LNJ     $B5,<GOBOT              REWIND, WAIT
002575  0A1C  D380 0FEC                          LNJ     $B5,<INSTAT             GET STATUS
002576  0A1E  9800 21B4                          LDR     $R1,<STAT1
002577  0A20  9970 8200                          CMR     $R1,=Z'8200'            RDY, BOT
002578  0A22  098F                              BNE     >FILEA                  B IF STAT1 NG
002579  0A23  9800 21B5                          LDR     $R1,<STAT2
002580  0A25  9630 20FA                          XOR     $R1,<DXDENS,$R3         MASK WITH DENSITY BIT
002581  0A27  A0B0 2102                          LDH     $R2,<DXMD,$R3           GET MODE
002582  0A29  A1F0 5220                          CMH     $R2,='R '
002583  0A2B  0983                              BNE     >FILEC                  B IF NOT MODE "R"
002584  0A2C  8851                              LBF     =$R1,=Z'2000'           IGNORE "PROTECT" STATUS BIT
        0A2D  2000
002585  0A2E  9970 8000                 FILEC   CMR     $R1,=Z'8000'
002586  0A30  0904                              BE      >FILEB                  B IF STAT 2 OK
002587  0A31  D380 1030                 FILEA   LNJ     $B5,<FNER               STAT NG AFTER RWND
002588  0A33  3134                     FNER14  DC      '14'                    LABEL2
002589                                  *
002590  0A34  D380 131A                 FILEB   LNJ     $B5,<WFM                WRITE A FILE MARK
002591  0A36  8386                              JMP     $B6                     RETURN TO HANDLER
002592                                  *****************************************************************
002593                                  * TEST AA, WRITE RECORD CONTAINING FILE ADDRESS, READ, CHECK
002594                                  *
002595  0A37  D380 0FA2                 AAST    LNJ     $B5,<LABEL
002596  0A39  4141                      AA      DC      'AA'                    LABEL1
002597                                  *
002598  0A3A  9B80 2306                          LAB     $B1,<SA10DV             DEVICE RUPTS AT LVL 10
002599  0A3C  9F80 000A                 X        STB     $B1,<ZHISAZ+10*$AF
002600  0A3E  9B80 17B3                          LAB     $B1,<LEVV10
002601  0A40  9F80 2309                          STB     $B1,<SA10P
002602                                  *
002603  0A42  9830 20FE                          LDR     $R1,<DXFN,$R3           GET CURRENT FILE NMBR
002604  0A44  1D08                              CMV     $R1,=8
002605  0A45  0387                              BLE     >AA2                    B IF NO RUPTS WANTED
002606                                  *
002607  0A46  8070 000A                 AA1     IO      =10,<OTRUPT
        0A48  0000 22BF
002608  0A4A  07FC                              B10F    >AA1
002609  0A4B  0F86                              B       >AA3                    CONTINUE
002610                                  *
002611  0A4C  8070 0000                 AA2     IO      =0,<OTRUPT              INHIBIT INTERRUPTS
        0A4E  0000 22BF
002612  0A50  07FC                              B10F    >AA2
002613                                  *
002614  0A51  D380 10DE                 AA3     LNJ     $B5,<WREC               WRITE RECORD
002615  0A53  0000                              RESV    1,0                     DATA (INSERTED IN WREC)
002616  0A54  0012                              DC      18                      BYTE RANGE
002617  0A55  17E6                              DC      <WBUF                   FROM...
002618  0A56  10C3                              DC      <RTS                    RETRY ROUTINE (BSR/BSR/FSR/ERA)
002619  0A57  0A51                              DC      <AA3                    RETRY LOCATION
002620  0A58  10DD                              DC      <ABR3                   ABORT ROUTINE (NOP)
002621  0A59  0A6F                              DC      <AB                     ABORT LOCATION (NEXT TEST)
002622                                  *
002623                                  * CHECK WHICH MODE, THEN READ AND CHECK DATA.
002624                                  *
002625  0A5A  D380 0FDB                          LNJ     $B5,<RWAQ               CHECK MODE
002626  0A5C  0F89                              B       >AA4                    R
002627  0A5D  0F92                              B       >AB                     W
002628  0A5E  0F7F                              NOP     >$-1                    A
002629  0A5F  8280 20E7                          LB      <AUFL,=1                Q, CHECK IF TEST AU
        0A61  0001
002630  0A62  0503                              BBT     >AA4                    B IF TEST AU
002631  0A63  D380 142B                          LNJ     $B5,<BSRW               BACK A RECORD AND WAIT
002632                                  *
002633  0A65  D380 10B6                 AA4     LNJ     $B5,<SET                SET UP FOR READ
002634                                  *
002635  0A67  D380 118D                 AA5     LNJ     $B5,<RREC               READ A RECORD
002636  0A69  0012                              DC      18                      BYTE RANGE
002637  0A6A  1C66                              DC      <RBUF                   TO HERE
002638  0A6B  10D1                              DC      <RT6                    RETRY ROUTINE (BSR/BSR/FSR)
002639  0A6C  0A67                              DC      <AA5                    RETRY LOCATION
002640  0A6D  10DD                              DC      <ABR3                   ABORT ROUTINE (NOP)
002641  0A6E  0A6F                              DC      <AB                     ABORT LOCATION (NEXT TEST)
002642                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002643                                  * TEST AB, WRITE RECORD OF ZERO'S, READ AND CHECK
002644                                  *
002645  0A6F  D380 0FA2                 AB      LNJ     $B5,<LABEL
002646  0A71  4142                              DC      'AB'                    LABEL1
002647  0A72  D380 11D0                 ABU     LNJ     $B5,<ONES               WRITE AND CHECK RECORD WITH...
002648  0A74  0000                              DC      X'00'                   DATA
002649  0A75  0113                              DC      275                     BYTE RANGE (DEC)
002650                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002651                                  * TEST AC, WRITE RECORD OF ONE'S IN BIT 7 OF EACH BYTE, READ AND CHECK.
002652                                  *
002653  0A76  D380 0FA2                 AC      LNJ     $B5,<LABEL
002654  0A78  4143                              DC      'AC'                    LABEL1
002655  0A79  D380 11D0                 ACU     LNJ     $B5,<ONES               WRITE AND CHECK RECORD WITH...
002656  0A7B  0001                              DC      X'01'                   DATA
002657  0A7C  076C                              DC      1900                    BYTE RANGE (DEC)
002658                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002659                                  * TEST AD, WRITE RECORD OF ONE'S IN BIT 6 OF EACH BYTE, READ AND CHECK.
002660                                  *
002661  0A7D  D380 0FA2                 AD      LNJ     $B5,<LABEL
002662  0A7F  4144                              DC      'AD'                    LABEL1
002663  0A80  D380 11D0                 ADU     LNJ     $B5,<ONES               WRITE AND CHECK RECORD WITH...
002664  0A82  0002                              DC      X'02'                   DATA
002665  0A83  06BF                              DC      1727                    BYTE RANGE (DEC)
002666                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002667                                  * TEST AE, WRITE RECORD OF ONE'S IN BIT 5 OF EACH BYTE, READ AND CHECK.
002668                                  *
002669  0A84  D380 0FA2                 AE      LNJ     $B5,<LABEL
002670  0A86  4145                              DC      'AE'                    LABEL1
002671  0A87  D380 11D0                 AEU     LNJ     $B5,<ONES               WRITE AND CHECK RECORD WITH...
002672  0A89  0004                              DC      X'04'                   DATA
```

```
002673  0A8A  03EC                      DC    1004                BYTE RANGE (DEC)
002674
002675                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002675                                  * TEST AF, WRITE RECORD OF ONE'S IN BIT 4 OF EACH BYTE, READ AND CHECK.
002676                                  *
002677  0A8B  0380 0FA2                 AF    LNJ   $B5,<LABEL
002678  0A8D  4146                            DC    'AF'                LABEL1
002679  0A8E  0380 11D0                 AFU   LNJ   $B5,<ONES             WRITE AND CHECK RECORD WITH...
002680  0A90  0008                            DC    X'08'               DATA
002681  0A91  0083                            DC    131                 BYTE RANGE (DEC)
002682                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002683                                  * TEST AG, WRITE RECORD OF ONE'S IN BIT 3 OF EACH BYTE, READ AND CHECK.
002684                                  *
002685  0A92  0380 0FA2                 AG    LNJ   $B5,<LABEL
002686  0A94  4147                            DC    'AG'                LABEL1
002687  0A95  0380 11D0                 AGU   LNJ   $B5,<ONES             WRITE AND CHECK RECORD WITH...
002688  0A97  0010                            DC    X'10'               DATA
002689  0A98  0324                            DC    804                 BYTE RANGE (DEC)
002690                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002691                                  * TEST AH, WRITE RECORD OF ONE'S IN BIT 2 OF EACH BYTE, READ AND CHECK.
002692                                  *
002693  0A99  0380 0FA2                 AH    LNJ   $B5,<LABEL
002694  0A9B  4148                            DC    'AH'                LABEL1
002695  0A9C  0380 11D0                 AHU   LNJ   $B5,<ONES             WRITE AND CHECK RECORD WITH...
002696  0A9E  0020                            DC    X'20'               DATA
002697  0A9F  059B                            DC    1435                BYTE RANGE (DEC)
002698                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002699                                  * TEST AI, WRITE RECORD OF ONE'S IN BIT 1 OF EACH BYTE, READ AND CHECK.
002700                                  *
002701  0AA0  0380 0FA2                 AI    LNJ   $B5,<LABEL
002702  0AA2  4149                            DC    'AI'                LABEL1
002703  0AA3  0380 11D0                 AIU   LNJ   $B5,<ONES             WRITE AND CHECK RECORD WITH...
002704  0AA5  0040                            DC    X'40'               DATA
002705  0AA6  0138                            DC    312                 BYTE RANGE (DEC)
002706                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002707                                  * TEST AJ, WRITE RECORD OF ONE'S IN BIT 0 OF EACH BYTE, READ AND CHECK.
002708                                  *
002709  0AA7  0380 0FA2                 AJ    LNJ   $B5,<LABEL
002710  0AA9  414A                            DC    'AJ'                LABEL1
002711  0AAA  0380 11D0                 AJU   LNJ   $B5,<ONES             WRITE AND CHECK RECORD WITH...
002712  0AAC  0080                            DC    X'80'               DATA
002713  0AAD  001D                            DC    29                  BYTE RANGE (DEC)
002714                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002715                                  * TEST AK, WRITE RECORD OF ONE'S, READ AND CHECK.
002716                                  *
002717  0AAE  0380 0FA2                 AK    LNJ   $B5,<LABEL
002718  0AB0  414B                            DC    'AK'                LABEL1
002719  0AB1  0380 11D0                 AKU   LNJ   $B5,<ONES             WRITE AND CHECK RECORD WITH...
002720  0AB3  00FF                            DC    X'FF'               DATA
002721  0AB4  07AF                            DC    1967                BYTE RANGE (DEC)
002722                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002723                                  * TEST AL, WRITE RECORD WITH CHECKER-BOARD PATTERN, READ AND CHECK.
002724                                  *
002725  0AB5  0380 0FA2                 AL    LNJ   $B5,<LABEL
002726  0AB7  414C                            DC    'AL'                LABEL1
002727  0AB8  0380 11D0                 ALU   LNJ   $B5,<ONES             WRITE AND CHECK RECORD WITH...
002728  0ABA  00A9                            DC    X'A9'               DATA
002729  0ABB  0198                            DC    408                 BYTE RANGE (DEC)
002730                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002731                                  * TEST AM, WRITE RECORD WITH CHECKER-BOARD PATTERN, READ AND CHECK.
002732                                  *
002733  0ABC  0380 0FA2                 AM    LNJ   $B5,<LABEL
002734  0ABE  414D                            DC    'AM'                LABEL1
002735  0ABF  0380 11D0                 AMU   LNJ   $B5,<ONES             WRITE AND CHECK RECORD WITH...
002736  0AC1  0056                            DC    X'56'               DATA
002737  0AC2  005C                            DC    92                  BYTE RANGE (DEC)
002738                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002739                                  * TEST AN, WRITE RANDOM LENGTH RECORD, RANDOM DATA, READ, CHECK
002740                                  *
002741  0AC3  0380 0FA2                 AN    LNJ   $B5,<LABEL
002742  0AC5  414E                            DC    'AN'                LABEL1
002743  0AC6  0380 12D2                 ANU   LNJ   $B5,<RLD              WRITE AND CHECK RECORD WITH RANDOM DATA
002744  0AC8  8280 20E7                       LB    <AUFL,=1
        0ACA  0001
002745  0ACD  0583                            BBF   >AO                 B IF NOT IN TEST AU
002746  0ACC  0380 0B9B                       LNJ   $B5,<AU6            RETURN TO TEST AU
002747                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002748                                  * TEST AO, REPEAT TEST AN WITH NEW RANGE AND DATA
002749                                  *
002750  0ACE  0380 0FA2                 AO    LNJ   $B5,<LABEL
002751  0AD0  414F                            DC    'AO'                LABEL1
002752  0AD1  0380 12D2                 AOU   LNJ   $B5,<RLD              WRITE, CHECK RECORD WITH RANDOM DATA
002753  0AD3  8280 20E7                       LB    <AUFL,=1
        0AD5  0001
002754  0AD6  0583                            BBF   >APGRS              B IF NOT IN TEST AU
002755  0AD7  0380 0B9B                       LNJ   $B5,<AU6            RETURN TO TEST AU
002756                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002757                                  * TESTS AP, AQ, AR, AS. DO AQ & AR FROM RAND BUFFERS ONLY IF G.T. 8K.
002758                                  *
002759  0AD9  9C80 0000          X     APGRS LDB   $B1,<ZV$HK            GET HMA IN B1
002760  0ADB  ABC0 2000          P           LAB   $B2,ZHCOMM+8192
002761  0ADD  9DD2                        CMB   $B1,=$B2
002762  0ADE  0300 0AE9                   BG    <APGRSA               B IF RAND BUFS TO BE USED (GT 8K)
002763                                  *
002764                                  * SET UP FOR STANDARD BUFFERS
002765                                  *
002766  0AE0  9B80 1C66                       LAB   $B1,<RBUF
002767  0AE2  9F80 0B1C                       STB   $B1,<AQC            STANDARD READ BUFFER
002768  0AE4  9B80 17E6                       LAB   $B1,<WBUF
002769  0AE6  9F80 0B36                       STB   $B1,<ARC            STANDARD WRITE BUFFER
002770  0AE8  0F8D                            B     >AP                 START TEST AP
002771                                  *
002772                                  * SET UP FOR RANDOM BUFFERS
002773                                  *
002774  0AE9  9BC1 FB80                 APGRSA LAB    $B1,$B1,-1152       ROOM AT TOP FOR 2K BYTES PLUS OVFL
002775  0AEB  9F80 218A                       STB   $B1,<MXAD
002776  0AED  0380 0FF7                       LNJ   $B5,<GRNADR         GET RANDOM ADDRESS IN B1
002777  0AEF  9F80 0B1C                       STB   $B1,<AQC            RANDOM READ BUFFER
002778  0AF1  0380 0FF7                       LNJ   $B5,<GRNADR         GET RANDOM ADDRESS IN B1
002779  0AF3  9F80 0B36                       STB   $B1,<ARC            RANDOM WRITE BUFFER
002780                                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002781                                  * TEST AP, WRITE RECORD OF RANDOM DATA FROM "WBUF".
002782                                  *
002783  0AF5  0380 0FA2                 AP    LNJ   $B5,<LABEL
```

```
002784  0AF7  4150                    DC      'AP'                    LABEL1
002785  0AF8  8280 20E7        APU     LB      <AUFL,=1
        0AFA  0001
002786  0AFB  0585                    BBF     >APA                    B IF NOT IN TEST AU
002787                          *
002788  0AFC  9B80 1C66                LAB     $B1,<RBUF
002789  0AFE  9F80 0B1C                STB     $B1,<AQC                USE STANDARD RBUF IF TEST AU
002790                          *
002791  0B00  D380 10DE        APA     LNJ     $B5,<WREC               WRITE RECORD
002792  0B02  5244                    DC      'RD'                    RANDOM DATA
002793  0B03  0400                    DC      1024                    BYTE RANGE
002794  0B04  17E6                    DC      <WBUF                   BUFFER ADDRESS
002795  0B05  10C3                    DC      <RT5                    RETRY ROUTINE (BSR/BSR/FSR/ERA)
002796  0B06  0B00                    DC      <APA                    RETRY LOCATION
002797  0B07  10DD                    DC      <ABR3                   ABORT ROUTINE (NOP)
002798  0B08  0B27                    DC      <AR                     ABORT LOCATION (SKIP NEXT TEST)
002799                          *
002800                          * CHECK FOR MODE
002801                          *
002802  0B09  D380 0FDB                LNJ     $B5,<RWAQ               MODE?
002803  0B0B  0F89                    B       >AQ                     R, GO ON AND DO READ
002804  0B0C  0F9B                    B       >AR                     W, SKIP NEXT READ
002805  0B0D  0F7F                    NOP     >$-1                    A
002806  0B0E  8280 20E7               LB      <AUFL,=1                Q, CHECK IF IN TEST AU
        0B10  0001
002807  0B11  0506                    BBT     >AQA                    B IF IN TEST AU
002808  0B12  D380 142B               LNJ     $B5,<BSRW               BACK A RECORD
002809                          *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002810                          * TEST AQ, READ PREVIOUS RECORD INTO RANDOM BUFFER AND CHECK.
002811                          * USE STANDARD BUFFER LOCATION IF IN TEST AU.
002812                          *
002813  0B14  D380 0FA2        AQ      LNJ     $B5,<LABEL
002814  0B16  4151                    DC      'AQ'                    LABEL1
002815                          *
002816  0B17  D380 10B6        AQA     LNJ     $B5,<SET                SET UP FOR READ
002817  0B19  D380 118D        AQB     LNJ     $B5,<RREC               READ A REC
002818  0B1B  0400                    DC      1024                    BYTE RANGE
002819  0B1C  0000              AQC     RESV    $AF,0                   BUFFER ADDRESS
002820  0B1D  10D1                    DC      <RT6                    RETRY ROUTINE (BSR/BSR/FSR)
002821  0B1E  0B19                    DC      <AQB                    RETRY LOCATION
002822  0B1F  10DD                    DC      <ABR3                   ABORT ROUTINE (NOP)
002823  0B20  0B21                    DC      <AQD                    ABORT LOCATION (END OF TEST)
002824                          *
002825  0B21  8280 20E7        AQD     LB      <AUFL,=1
        0B23  0001
002826  0B24  0583                    BBF     >AR                     B IF NOT TEST AU
002827  0B25  D380 0B9B               LNJ     $B5,<AU6                RETURN TO TEST AU
002828                          *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002829                          * TEST AR, WRITE RECORD OF RANDOM DATA FROM RANDOM LOCATION.
002830                          * USE "WBUF" IF ONLY 8K.
002831                          *
002832  0B27  D380 0FA2        AR      LNJ     $B5,<LABEL
002833  0B29  4152                    DC      'AR'                    LABEL1
002834  0B2A  8280 20E7        ARU     LB      <AUFL,=1
        0B2C  0001
002835  0B2D  0585                    BBF     >ARA                    B IF NOT AU
002836  0B2E  9B80 17E6               LAB     $B1,<WBUF
002837  0B30  9F80 0B36               STB     $B1,<ARC                USE STANDARD BUFFER
002838                          *
002839  0B32  D380 10DE        ARA     LNJ     $B5,<WREC               WRITE
002840  0B34  5244                    DC      'RD'                    RAND DATA
002841  0B35  0400                    DC      1024                    BYTE RANGE
002842  0B36  0000              ARC     RESV    $AF,0                   BUFFER ADDRESS
002843  0B37  10C3                    DC      <RT5                    RETRY ROUTINE (BSR/BSR/FSR/ERA)
002844  0B38  0B32                    DC      <ARA                    RETRY LOCATION
002845  0B39  10DD                    DC      <ABR3                   ABORT ROUTINE (NOP)
002846  0B3A  0B54                    DC      <ASC                    ABORT LOCATION (SKIP NEXT TEST)
002847                          *
002848                          * CHECK FOR MODE
002849                          *
002850  0B3B  D380 0FDB                LNJ     $B5,<RWAQ               MODE?
002851  0B3D  0F8A                    B       >AS                     R, CONTINUE WITH READ
002852  0B3E  0F96                    B       >ASC                    W, SKIP THE READ
002853  0B3F  0F7F                    NOP     >$-1                    A
002854  0B40  8280 20E7               LB      <AUFL,=1                Q, CHECK IF IN TEST AU
        0B42  0001
002855  0B43  0500 0B4A               BBT     <ASA                    B IF AU (DON'T CHANGE LABEL1 = "AU")
002856  0B45  D380 142B               LNJ     $B5,<BSRW               BACKSPACE FOR THE FOLLOWING READ
002857                          *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002858                          * TEST AS, READ PREVIOUS RECORD INTO "RBUF" AND CHECK DATA.
002859                          *
002860  0B47  D380 0FA2        AS      LNJ     $B5,<LABEL
002861  0B49  4153                    DC      'AS'                    LABEL1
002862  0B4A  D380 10B6        ASA     LNJ     $B5,<SET                SET UP FOR READ
002863  0B4C  D380 118D        ASB     LNJ     $B5,<RREC               READ
002864  0B4E  0400                    DC      1024                    BYTE RANGE
002865  0B4F  1C66                    DC      <RBUF                   BUF ADR
002866  0B50  10D1                    DC      <RT6                    RETRY ROUTINE (BSR/BSR/FSR)
002867  0B51  0B4C                    DC      <ASB                    RETRY LOCATION
002868  0B52  10DD                    DC      <ABR3                   ABORT ROUTINE (NOP)
002869  0B53  0B54                    DC      <ASC                    ABORT LOCATION
002870                          *
002871  0B54  8280 20E7        ASC     LB      <AUFL,=1
        0B56  0001
002872  0B57  0583                    BBF     >AT                     B IF NOT TEST AU
002873  0B58  D380 0B9B               LNJ     $B5,<AU6                RETURN TO TEST AU
002874                          *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002875                          * WRITE (OR SPACE) FILE MARK AT END OF DATA FILE
002876                          *
002877  0B5A  D380 0FA2        AT      LNJ     $B5,<LABEL
002878  0B5C  4154                    DC      'AT'                    LABEL1
002879  0B5D  D380 0FDB               LNJ     $B5,<RWAQ               MODE?
002880  0B5F  0F86                    B       >ATA                    R, READ FM & SKIP AU TEST
002881  0B60  0F85                    B       >ATA                    W, WRITE FM & SKIP AU TEST
002882  0B61  0F7F                    NOP     >$-1                    A WRITE FM, THEN DO AU TEST
002883  0B62  D380 131A               LNJ     $B5,<WFM                Q, DITTO
002884  0B64  0F85                    B       >AU                     A/Q, CONTINUE WITH RANDOM SEEK TEST
002885                          *
002886  0B65  D380 131A        ATA     LNJ     $B5,<WFM                R/W, WRITE (READ) FILE MARK
002887  0B67  0F80 0BB1               B       <AU7                    GO TO END OF TEST AU
002888                          *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002889                          * TEST AU, ARBITRARILY SKIP AROUND WITHIN FILE & CHECK DATA
002890                          *
```

```
002891  0B69  D380 0FA2          AU      LNJ     $B5,<LABEL
002892  0B6B  4155                      DC      'AU'                    LABEL1
002893  0B6C  D380 13F5                  LNJ     $B5,<BSFW               BACK A FILE
002894  0B6E  D380 13F5                  LNJ     $B5,<BSFW               DITTO
002895  0B70  D380 139E                  LNJ     $B5,<FSFEOT             FORWARD A FILE, PAST EOT IF NECESSARY
002896                           *
002897  0B72  8900 20E7                  LBT     <AUFL,=1                SET AU TEST FLAG
        0B74  0001
002898                                   CALL    <V$FR,RANDOM,DC8        GET SOME RANDOM NUMBERS
        0B75  FBC0 0003
        0B77  D380 0000          X
        0B79  0F80
        0B7A  2195
        0B7B  20F1
002899                           *
002900                           * NOW DETERMINE HOW TO GET TO RANDOM RECORD NUMBERS (R5 = AU CNTR)
002901                           *
002902  0B7C  1CF8                      LDV     $R1,=-8
002903  0B7D  9F00 20EB                  STR     $R1,<CNT8               CNTR FOR 8 SPACEING OPERATIONS
002904  0B7F  AB80 2195                  LAB     $B2,<RANDOM             PNTR TO RANDOM TABLE
002905  0B81  0B72              AU1     LDR     $R5,+$B2                GET A NMBR, BUMP FOR NEXT ONE
002906  0B82  0570 000F                  AND     $R5,=X'F'               TRUNCATE TO DECIMAL 15
002907  0B84  8AD5                      INC     =$R5                    MAKE IT BETWEEN 1-16 DEC
002908  0B85  0230 210E                  SUB     $R5,<DXRN.$R3           DIFFERENCE FROM CURRENT POSITION
002909  0B87  5880 0B8E                  BGEZ    $R5,<AU3                B IF NECESSARY TO FORWARD SPACE
002910                           *
002911                           * IT IS NECESSARY TO BACKSPACE TO GET THERE
002912                           *
002913  0B89  D380 142B          AU2     LNJ     $B5,<BSRW               BACKSPACE A RECORD
002914  0B8B  57FE                      BINC    $R5,>AU2                BACK SOME MORE
002915  0B8C  0F80 0B94                  B       <AU5                    WE'VE ARRIVED
002916                           *
002917                           * IT'S NECESSARY TO FORWARD SPACE TO GET THERE
002918                           *
002919  0B8E  8255               AU3     NEG     =$R5                    MAKE IT NEGATIVE
002920  0B8F  5900 0B94                  BEZ     $R5,<AU5                TAPE ALREADY IN POSITION
002921  0B91  D380 13C1          AU4     LNJ     $B5,<FSRW               FORWARD SPACE A RECORD
002922  0B93  57FE                      BINC    $R5,>AU4                FORWARD SPACE SOME MORE
002923                           *
002924                           * WE'VE ARRIVED, PREPARE TO RECONSTRUCT DATA AND READ IT.
002925                           *
002926  0B94  9B60 2333          AU5     LAB     $B1,<LIST               LIST OF POSSIBLE AU TESTS
002927  0B96  9830 210E                  LDR     $R1,<DXRN.$R3           CURRENT RECORD NMBR
002928  0B98  88D1                      DEC     =$R1                    ("LIST" STARTS WITH SECOND RECORD)
002929  0B99  BC91                      LDB     $B3,$B1.$R1             FORM ADDRESS TO TEST
002930  0B9A  8383                      JMP     $B3                     DO THE TEST
002931                           *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002932                           * AFTER DOING THE READ PORTION OF TEST, RETURN HERE AND CHECK MODE
002933                           *
002934  0B9B  0F80 0BA6          AU6     STB     $B5,<AU6B5              SAVE POSSIBLE RETURN ADDRESS
002935  0B9D  D380 0FDB                  LNJ     $B5,<RWAQ               MODE?
002936  0B9F  0F7F                       NOP     >$-1                    R, NO AU TEST, RETURN
002937  0BA0  0F83                       B       >AU6A                   W, DITTO
002938  0BA1  0F7F                       NOP     >$-1                    A, CONTINUE W/NEXT REC OF AU TEST
002939  0BA2  0F85                       B       >AU6C                   Q, DITTO
002940                           *
002941  0BA3  DC80 0BA6          AU6A    LDB     $B5,<AU6B5              R/W, SET UP FOR RETURN
002942  0BA5  8385                      JMP     $B5                     RETURN TO TEST "A" IN LINE
002943  0BA6  0000               AU6B5   RESV    $AF,0                   RETURN ADDRESS
002944                           *
002945  0BA7  8A80 20EB          AU6C    INC     <CNT8                   A/Q, BUMP CNTR FOR 8 RECORDS
002946  0BA9  8980 20EB                  CMZ     <CNT8                   CHECK CNTR
002947  0BAB  0800 0B81                  BAL     <AU1                    LOOP BACK FOR NEXT RECORD
002948  0BAD  8700 20E7                  CL      <AUFL                   RESET AU FLAG, TEST AU DONE
002949  0BAF  D380 139E                  LNJ     $B5,<FSFEOT             GO AHEAD FOR NEXT FILE
002950                           *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002951                           * AU TEST DONE, CHECK FOR E-O-T OR END-PASS
002952                           *
002953  0BB1  8800 2112          AU7     LBF     <EOTFLG,=1              CHECK AND CLEAR EOT FLAG
        0BB3  0001
002954  0BB4  0500 0BD4                  BBT     <NDPS                   B IF EOT, REPORT END OF PASS
002955  0BB6  D380 0FDB                  LNJ     $B5,<RWAQ               MODE?
002956  0BB8  0F7F                       NOP     >$-1                    R, COMPARE "FILES"
002957  0BB9  0F7F                       NOP     >$-1                    W, DITTO
002958  0BBA  0F88                       B       >AU8                    A, DITTO
002959                           *
002960  0BBB  9830 20FE                  LDR     $R1,<DXFN.$R3           Q, CHECK FOR 16 FILES
002961  0BBD  1D11                       CMV     $R1,=17                 16 FILES = 17 FILE MARKS
002962  0BBE  0900 0BD4                  BE      <NDPS                   REPORT END OF PASS
002963  0BC0  0F80 0BCD                  B       <AU9                    NOT 16 FILES YET, CONTINUE
002964                           *
002965  0BC2  8980 217F          AU8     CMZ     <FILES
002966  0BC4  0900 0BCD                  BE      <AU9                    B IF NO LIMIT, GO TO EOT
002967  0BC6  9830 20FE                  LDR     $R1,<DXFN.$R3           GET FILE COUNT
002968  0BC8  88D1                       DEC     =$R1
002969  0BC9  9900 217F                  CMR     $R1,<FILES              "FILES" REACHED?
002970  0BCB  0280 0BD4                  BGE     <NDPS                   B IF PASS
002971                           *
002972  0BCD  9CB0 210A          AU9     LDB     $B1,<DXPT.$R3
002973  0BCF  9BC1 FFFF                  LAB     $B1,$B1.-$AF            NEXT TIME AROUND, REPEAT AA-AT,AU
002974  0BD1  9FB0 210A                  STB     $B1,<DXPT.$R3
002975  0BD3  8386                       JMP     $B6                     RETURN TO HANDLER
002976                           *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
002977                           * REPORT END-OF-PASS MESSAGE
002978                           *
002979                           * FIRST WRITE END-PASS RECORD ON TAPE
002980                           *
002981  0BD4  D380 0FA2          NDPS    LNJ     $B5,<LABEL
002982  0BD6  4550               EP      DC      'EP'                    LABEL1
002983                           *
002984  0BD7  8070 0000          EP4     IO      =0,<OTRUPT              INHIBIT INTERRUPTS
        0BD9  0000 22BF
002985  0BDB  07FC                       BIOF    >EP4
002986                           *
002987  0BDC  D380 0FDB                  LNJ     $B5,<RWAQ               MODE?
002988  0BDE  0F87                       B       >EP1                    R, NO WRITING
002989  0BDF  0F7F                       NOP     >$-1                    W
002990  0BE0  0F7F                       NOP     >$-1                    A
002991                           *
002992  0BE1  D380 11D0                  LNJ     $B5,<ONES               Q, WRITE RECORD
002993  0BE3  00FF                       DC      X'FF'                   DATA
002994  0BE4  0014                       DC      20                      BYTE RANGE
002995                           *
```

```
002996                                  * NOW REPORT MODE
002997                                  *
002998                                  EP1     CALL    ZV$1.ZV$TC,MSMODE       PRINT "MODE"
        0BE5  FBC0 0003
        0BE7  D380 0000         X
        0BE9  0F80
        0BEA  225F
002999  0BEB  9830 2102                      LDR     $R1,<DXMD.$R3           GET MODE (ASCII)
003000  0BED  D380 0FCB                      LNJ     $B5,<VDTR               PRINT IT
003001                                  *
003002                                  * PASS COUNT (DECIMAL)
003003                                  *
003004                                          CALL    ZV$1,MSPASS             PRINT "PASS"
        0BEF  FBC0 0003
        0BF1  D380 0000         X
        0BF3  0F80
        0BF4  2277
003005  0BF5  9830 2106                      LDR     $R1,<DXPS.$R3           GET PASS COUNT
003006  0BF7  9F00 21B7                      STR     $R1,<TEMPA
003007                                          CALL    ZV$1H.ZV$TD,TEMPA       PRINT PASS COUNT
        0BF9  FBC0 0003
        0BFB  D380 0000         X
        0BFD  0F80
        0BFE  21B7
003008                                  *
003009                                  * ERROR COUNT (DECIMAL)
003010                                  *
003011  0BFF  D380 145F                      LNJ     $B5,<BLANKS             PRINT SOME BLANKS
003012                                          CALL    ZV$1H.ZV$TD,ERCT        PRINT TOTAL ERROR COUNT
        0C01  FBC0 0003
        0C03  D380 0000         X
        0C05  0F80
        0C06  2113
003013                                          CALL    ZV$1,MSERR              PRINT "ERRS"
        0C07  FBC0 0003
        0C09  D380 0000         X
        0C0B  0F80
        0C0C  2239
003014                                  *
003015                                  * FILE COUNT (HEX)
003016                                  *
003017  0C0D  9830 20FE                      LDR     $R1,<DXFN.$R3           GET FILE COUNT
003018  0C0F  88D1                          DEC     =$R1                    FILES = FILE MARKS - 1
003019  0C10  9F00 21B7                      STR     $R1,<TEMPA
003020                                          CALL    ZV$1H,TEMPA             PRINT HEX FILE COUNT
        0C12  FBC0 0003
        0C14  D380 0000         X
        0C16  0F80
        0C17  21B7
003021                                          CALL    ZV$1,MSFILE             PRINT "HEX FILE(S)..."
        0C18  FBC0 0003
        0C1A  D380 0000         X
        0C1C  0F80
        0C1D  223F
003022                                  *
003023  0C1E  D380 1628                      LNJ     $B5,<CRLF               DO A LINE-FEED
003024  0C20  8700 2112                      CL      <EOTFLG                 CLEAR EOT FLAG
003025                                  *
003026                                  * NOW REWIND AND PREPARE TO START NEXT PASS
003027                                  *
003028  0C22  8000 2299                      IO      <REWIND,<OTTASK         REWIND
        0C24  0000 22C2
003029  0C26  0704                          BIOT    >EP2
003030  0C27  D380 1030                      LNJ     $B5,<FNER               SHOULD HAVE BEEN READY FOR REWIND
003031  0C29  3137                      FNER17 DC      '17'                    LABEL2
003032                                  *
003033  0C2A  8730 20FE                  EP2     CL      <DXFN.$R3               RESET FILE COUNT
003034  0C2C  8730 210E                      CL      <DXRN.$R3               RESET RECORD COUNT
003035  0C2E  8AB0 2106                      INC     <DXPS.$R3               INCREMENT PASS COUNT
003036  0C30  D380 0FDB                      LNJ     $B5,<RWAQ               MODES?
003037  0C32  8386                          JMP     $B6                     R, RETURN TO HANDLER
003038  0C33  0F83                          B       >EP3                    W, CANCEL MODE "W", ONE PASS ONLY
003039  0C34  8386                          JMP     $B6                     A, RETURN TO HANDLER
003040  0C35  8386                          JMP     $B6                     Q, DITTO
003041                                  *
003042  0C36  9870 2D30                  EP3     LDR     $R1,='-0'               GET "NULL" MODE
003043  0C38  9A53                          ADD     $R1,=$R3                MODIFY TO CURRENT UNIT
003044  0C39  9F30 2102                      STR     $R1,<DXMD.$R3           CANCEL MODE "WN"
003045  0C3B  8386                          JMP     $B6                     RETURN TO HANDLER
003046                                  ********************************************************************
003047                                  *
003048                                  * NEXT FOLLOWS THE DEBUG MODE
003049                                  *
003050                                  ********************************************************************
```

```
003051                          /*****************************************************************
003052                          * DEBUG MODE. PRIOR TO CONFIGURING CURRENT CHANNEL ADDRESS, MAKE
003053                          * SURE THAT THE INTERRUPTED UNIT IS INITIALIZED AND IN NORMAL MODE.
003054                          *
003055  0C3C 8070 0000          MODED   IO      =0,<OTRUPT              INHIBIT INTERRUPTS
       0C3E 0000 22BF
003056  0C40 D380 0FA2          D1      LNJ     $B5,<LABEL              SET MAJOR ERROR LABEL (SEE "SENSE")
003057  0C42 4431                       DC      'D1'                    LABEL1
003058  0C43 8800 20EA                  LDR     $R3,<DITE               GET INDEX
003059  0C45 9870 2D30                  LDR     $R1,='-0'
003060  0C47 9A53                       ADD     $R1,=$R3                ADD INDEX VALUE
003061  0C48 9F30 2102                  STR     $R1,<DXMD.$R3           CLEAR CURRENT MODE
003062                          *
003063  0C4A 8000 22A9          MODEDA  IO      <NORMAL,<OTCONF         RESET CONFIGURATION ON INTERRUPTED DRIVE
       0C4C 0000 22BD
003064  0C4E 07FC                       BIOF    >MODEDA
003065                          *
003066                          * CLEAR UP ANY SCREWY CONDITION THAT MAY EXIST IN PREVIOUS DRIVE.
003067                          *
003068  0C4F 9800 20F5                  LDR     $R1,<DRIVE
003069  0C51 1047                       SOR     $R1,7
003070                          *
003071                          * SET UP TEST MODE
003072                          *
003073  0C52 8000 22B3                  IO      <TESTMD,<OTCONT         (SET TEST MODE)
       0C54 0000 22BE
003074  0C56 8010 22AC                  IO      <LCHNO,$R1,<OTCONT      (OUTPUT LOGIC. CHAN NMBR)
       0C58 0000 22BE
003075  0C5A 8000 22AA                  IO      <INDEX,<OTCONT          (SET INDEX MODE)
       0C5C 0000 22BE
003076                          *
003077                          * INITIALIZE OLD CHANNEL. IS DIFFERENT FOR BDC2 AND BDC3.
003078                          *
003079  0C5E 8260 20E9                  LB      <BDCFLG,=1
       0C60 0001
003080  0C61 0580 0C79                  BBF     <MODEDB                 B IF BDC2 FIRMWARE
003081                          *
003082                          * BDC3 FIRMWARE, INITIALIZE ADAPTER AND DRU
003083                          *
003084  0C63 8070 9C03                  IO      =Z'9C03',<OTCONT        (LCN AAD4 CNST(03#))
       0C65 0000 22BE
003085  0C67 8070 8883                  IO      =Z'8883',<OTCONT        (ACCESS (FWCI))
       0C69 0000 22BE
003086  0C6B 8070 0008                  IO      =Z'0008',<OTCONT        (RESET DEVICE ADAPTER)
       0C6D 0000 22BE
003087  0C6F 8070 20B4                  IO      =Z'20B4',<OTCONT        (AC13)
       0C71 0000 22BE
003088  0C73 8070 7D6A                  IO      =Z'7D6A',<OTCONT        ((FWCI) TO ADAPTER)
       0C75 0000 22BE
003089                          *
003090  0C77 0F80 0C89                  B       <MODEDE                 INITIALIZE IS DONE
003091                          *
003092                          * BDC2 FIRMWARE, INITIALIZE.
003093                          *
003094  0C79 8070 0008          MODEDB  IO      =Z'0008',<OTCONT        (RESET DEVICE ADAPTER)
       0C7B 0000 22BE
003095  0C7D 8070 8883                  IO      =Z'8883',<OTCONT        (ACCESS (FWCI))
       0C7F 0000 22BE
003096  0C81 8070 20F4                  IO      =Z'20F4',<OTCONT        (AC123)
       0C83 0000 22BE
003097  0C85 8070 7D6A                  IO      =Z'7D6A',<OTCONT        ((FWCI) TO ADAPTER)
       0C87 0000 22BE
003098                          *
003099                          * CLEAR TEST MODE
003100                          *
003101  0C89 8000 22B1          MODEDE  IO      <SCRDY,<OTCONT          (SET CHAN RDY)
       0C8B 0000 22BE
003102  0C8D 8000 22B4                  IO      <WAITLP,<OTCONT         (B TO WAIT LOOP)
       0C8F 0000 22BE
003103  0C91 8000 22B0                  IO      <NOTEST,<OTCONT         (EXIT TEST MODE)
       0C93 0000 22BE
003104                          *
003105                          *
003106                          *
003107  0C95 9830 20F6                  LDR     $R1,<DRIVE0.$R3
003108  0C97 9F00 20F5                  STR     $R1,<DRIVE              SET DRIVE ADR
003109  0C99 D380 1341                  LNJ     $B5,<SETCHN             SET UP CHANNEL ADDRESSES
003110  0C9B FBC0 0003         MODEDJ  CALL    ZV$I.ZV$GC,MSPARA       (ASK FOR RANGE, DATA)
       0C9D D380 0000   X
       0C9F 0F80
       0CA0 226A
003111                          *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
003112                          * RANGE (1-800 HEX BYTES MAX), DATA PATTERN (00-FF HEX)
003113                          *
003114  0CA1 8800 219F                  LBF     <RNGFLG,=1              TEST OLD FLAG
       0CA3 0001
003115  0CA4 0583                       BBF     >MODEDK
003116  0CA5 8700 219E                  CL      <RNG
003117  0CA7 E800 219E          MODEDK  LDR     $R6,<RNG                GET OLD RANGE FOR DEFAULT
003118  0CA9 F800 2190                  LDR     $R7,<PTRN               GET OLD PATTERN FOR DEFAULT
003119  0CAB 8D00 21B7                  SDI     <TEMPA                  TRANSFER BOTH TO TEMPA, TEMPB
003120  0CAD FBC0 0003                  CALL    ZV$IH,TEMPA,DC2
       0CAF D380 0000   X
       0CB1 0F80
       0CB2 21B7
       0CB3 20ED
003121  0CB4 8C80 21B7                  LDI     <TEMPA                  GET BACK INTO R6, R7
003122  0CB6 6985                       BNEZ    $R6,>MODEDL             B IF RANGE NOT "ZER
003123  0CB7 E870 0800                  LDR     $R6,=X'800'             DEFAULT TO '800
003124  0CB9 8A80 219F                  INC     <RNGFLG                 SET FLAG, RANGE IS "ZERO"
003125  0CBB 6800 0C9B          MODEDL  BLZ     $R6,<MODEDJ             B IF RANGE TOO SMALL
003126  0CBD E970 0800                  CMR     $R6,=X'800'
003127  0CBF 0A00 0C9B                  BAG     <MODEDJ                 START OVER
003128  0CC1 EF00 219E                  STR     $R6,<RNG                SAVE NEW (OLD) RANGE
003129  0CC3 FF00 2190                  STR     $R7,<PTRN               SAVE NEW (OLD) DATA PATTERN
003130  0CC5 8800 2190                  LBF     <PTRN,=Z'FF00'          CLEAR LEFT BYTE OF DATA
       0CC7 FF00
003131                          *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
003132                          * ASSEMBLE LINK TABLE
003133                          *
003134                          MODEDC CALL   ZV$I.ZV$GC,MS_INK         ASK FOR LINKS
```

```
         0CC8  FBC0 0003
         0CCA  D380 0000       X
         0CCC  0F80
         0CCD  225B
003135   0CCE  1CEB                    LDV     $R1,=-21                ROOM FOR 20 LINKS PLUS "C/R"
003136   0CCF  9B80 22E5               LAB     $B1,<LINKS+21*$AF
003137   0CD1  AB80 22E4               LAB     $B2,<LNKTAB-$AF         (-$AF BECAUSE ASCII A IS 41, NOT 40)
003138                         MODEDD CALL    ZV$1A,ZVSTAT,TEMPA,DC3
         0CD3  FBC0 0003
         0CD5  D380 0000       X
         0CD7  0F80
         0CD8  21C0
         0CD9  21B7
         0CDA  20EF
003139   0CDB  8780 21B7               CLH     <TEMPA
003140   0CDD  A800 21B7               LDR     $R2,<TEMPA              GET LINK
003141   0CDF  2B0D                    CMV     $R2,=X'0D'              "C/R"?
003142   0CE0  0980 0CEB               BNE     <MODEDF                 B IF NOT C/R
003143   0CE2  1DEB                    CMV     $R1,=-21
003144   0CE3  0900 0CFD               BE      <MODEDG                 B IF C/R IS FIRST LINK
003145   0CE5  1900 0CC8               BEZ     $R1,<MODEDC             TOO MANY LINKS
003146   0CE7  CB80 0CFD               LAB     $B4,<MODEDG             EXECUTION LOCATION
003147   0CE9  CFDD                    STB     $B4,$B1.+$R1            SAVE LAST LINK BACK TO "MODEDG"
003148   0CEA  8384                    JMP     $B4                     GO ON AND EXECUTE
003149                         *
003150   0CEB  8780 21B7       MODEDF  CLH     <TEMPA
003151   0CED  A800 21B7               LDR     $R2,<TEMPA              GET A LINK
003152   0CEF  2B41                    CMV     $R2,=X'41'
003153   0CF0  0200 0CC8               BL      <MODEDC                 LESS THAN "A"?
003154   0CF2  2B5A                    CMV     $R2,=X'5A'
003155   0CF3  0300 0CC8               BG      <MODEDC                 GREATER THAN "Z"?
003156   0CF5  A570 001F               AND     $R2,=Z'001F'            FORM INDEX
003157   0CF7  1900 0CC8               BEZ     $R1,<MODEDC             RAN OUT OF LINK TABLE SPACE
003158   0CF9  CCA2                    LDB     $B4,$B2.$R2             GET ENTRY FROM LNKTAB
003159   0CFA  CFDD                    STB     $B4,$B1.+$R1            SAVE THE LINK
003160   0CFB  0F80 0CD3               B       <MODEDD                 GET ANOTHER LINK
003161                         *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
003162                         * "C/R" DETECTED, EXECUTE
003163                         *
003164   0CFD  8800 2112       MODEDG  LBF     <EOTFLG,=Z'0001'
         0CFF  0001
003165   0D00  0580 0D09               BBF     <MODEDH                 B IF NOT E-O-T
003166                                 CALL    ZV$1.ZV$TC,MSEOT        REPORT E-O-T
         0D02  FBC0 0003
         0D04  D380 0000       X
         0D06  0F80
         0D07  2235
003167   0D08  0000                    HLT                             HIT "EXECUTE" TO CONTINUE
003168                         *
003169   0D09  CB80 22D0       MODEDH  LAB     $B4,<LINKS
003170   0D0B  ECF4            MODED1  LDB     $B6,+$B4                FORM LINK TO ROUTINE
003171   0D0C  8386                    JMP     $B6                     DO THE ROUTINE
003172                         ********************************************************************************
```

```
003173                          /*****************************************************************
003174                          * THE FOLLOWING IS A LIST OF AVAILABLE SUBROUTINE LINKS.
003175                          *
003176                          *
003177                          *       A - PRINT CONTENTS OF WRITE BUFFER
003178                          *       B - PRINT CONTENTS OF READ BUFFER
003179                          *       F - FILL WRITE BUFFER, FIXED DATA
003180                          *       G - FILL WRITE BUFFER, RANDOM DATA
003181                          *       I - FILL WRITE BUFFER, ASCENDING COUNT
003182                          *       Q - TRANSFER READ BUFFER TO WRITE BUFFER
003183                          *       U - COMPARE READ BUFFER TO WRITE BUFFER
003184                          *
003185                          *       J - ESTABLISH RANDOM WRITE BUFFER UP TO 64K
003186                          *       K - ESTABLISH RANDOM READ BUFFER UP TO 64K
003187                          *       N - GENERATE RANDOM PARAMETERS (LENGTH, DATA)
003188                          *       P - PRINT PARAMETER SET (UNIT LENGTH DATA W/BUF R/BUF)
003189                          *
003190                          *       C - INITIALIZE
003191                          *       D - REWIND TO "B-O-T"
003192                          *       E - ERASE A BLOCK OF TAPE
003193                          *       L - SPARE SUBROUTINES (NOP'S)
003194                          *       X - EXIT TO "NEXT ?;"
003195                          *
003196                          *       M - WRITE A TAPE MARK
003197                          *       Y - FORWARD-SPACE A FILE
003198                          *       U - BACK-SPACE A FILE
003199                          *
003200                          *       W - WRITE A RECORD
003201                          *       R - READ A RECORD (DON'T CHECK DATA)
003202                          *       V - FORWARD-SPACE A RECORD
003203                          *       T - BACK-SPACE A RECORD
003204                          *
003205                          *       S - INPUT BOTH STATUS WORDS
003206                          *       Z - PRINT UNIT, FILE, RECORD, REMAINING RANGE, STATUS.
003207                          *
003208                          *****************************************************************
003209                          * ALL ROUTINES EXCEPT "H" WILL RETURN TO "MODEDI" UPON COMPLETION
003210                          *****************************************************************
003211                          * PRINT CONTENTS OF WRITE BUFFER
003212                          *
003213  0D0D  9C80 2184         AAX     LDB     $B1,<JJPT               WRITE BUFFER POINTER
003214  0D0F  8751                      CL      =$R1                    R1 INDEXES THROUGH BUFFER BY BYTES
003215  0D10  2CF0              AAA     LDV     $R2,=-16                COUNT 16 BYTES PER LINE
003216  0D11  C870 8020                 LDR     $R4,=Z'8020'            CR/LF AND LEADING SPACE
003217  0D13  CF00 21B8                 STR     $R4,<TEMPB              CONTROL WORD
003218  0D15  9F00 21B7                 STR     $R1,<TEMPA              SAVE BYTE NUMBER
003219                                  CALL    ZV$TH,TEMPA,TEMPB       PRINT BYTE NUMBER
        0D17  FBC0 0003
        0D19  D380 0000         X
        0D1B  0F80
        0D1C  21B7
        0D1D  21B8
003220  0D1E  9F57                      STR     $R1,=$R7                SAVE BYTE INDEX
003221  0D1F  9870 2D2D                 LDR     $R1,='--'
003222  0D21  D380 0FCB                 LNJ     $B5,<VDTR               PRINT DASHES
003223  0D23  9857                      LDR     $R1,=$R7                RESTORE BYTE INDEX
003224                          *
003225  0D24  C2DD              AAB     LLH     $R4,$B1,+$R1            GET DATA
003226  0D25  CF00 21B7                 STR     $R4,<TEMPA
003227                                  CALL    ZV$HA,TEMPA,VRBL        CONVERT TO ASCII
        0D27  FBC0 0003
        0D29  D380 0000         X
        0D2B  0F80
        0D2C  21B7
        0D2D  21BB
003228  0D2E  C800 21BD                 LDR     $R4,<VRBL+2             GET ASCII
003229  0D30  CF00 0D48                 STR     $R4,<AAD+1              SAVE AS PART OF MESSAGE
003230                                  CALL    ZV$T,AAD                PRINT " XX"
        0D32  FBC0 0003
        0D34  D380 0000         X
        0D36  0F80
        0D37  0D47
003231  0D38  9900 219E                 CMR     $R1,<RNG
003232  0D3A  0287                      BGE     >AAC                    RETURN
003233  0D3B  2780 0D24                 BINC    $R2,<AAB
003234  0D3D  D380 100F                 LNJ     $B5,<SENSE              CHECK FOR KEYBOARD ACTIVITY
003235  0D3F  0F80 0D10                 B       <AAA
003236  0D41  D380 1628         AAC     LNJ     $B5,<CRLF               DO A LINE FEED
003237  0D43  D380 100F                 LNJ     $B5,<SENSE              CHECK FOR KEYBOARD ACTIVITY
003238  0D45  0F80 0D0B                 B       <MODEDI                 RETURN FOR NEXT DEBUG LINK
003239                          *
003240  0D47  5C20              AAD     DC      '\ '                    START OF MESSAGE (ONE BLANK)
003241  0D48  0000                      RESV    1,0                     SPACE FOR DATA
003242  0D49  2424                      DC      '$$'                    END OF MESSAGE
003243                          *****************************************************************
003244                          * PRINT CONTENTS OF READ BUFFER
003245                          *
003246  0D4A  9C80 2185         BB      LDB     $B1,<KKPT               READ BUFFER POINTER
003247  0D4C  8751                      CL      =$R1                    INDEX THROUGH BUFFER BY BYTES
003248  0D4D  0F80 0D10                 B       <AAA                    CONTINUE AS WITH LINK "AA"
003249                          *****************************************************************
003250                          * INITIALIZE
003251                          *
003252  0D4F  8000 22AB         CC      IO      <INTBDC,<OTCONT         INITIALIZE THE BDC
        0D51  0000 22BE
003253  0D53  D380 0FEC                 LNJ     $B5,<INSTAT             WAIT FOR N/BSY
003254  0D55  D380 100F                 LNJ     $B5,<SENSE              CHECK FOR KEYBOARD ACTIVITY
003255  0D57  0F80 0D0B                 B       <MODEDI                 GO BACK FOR NEXT DEBUG LINK
003256                          *****************************************************************
003257                          * COMPARE READ BUFFER TO WRITE BUFFER
003258                          *
003259  0D59  D380 0FA2         DD      LNJ     $B5,<LABEL              SET MAJOR ERROR LABEL
003260  0D5B  4444                      DC      'DD'                    LABEL1
003261  0D5C  9B80 0D6E                 LAB     $B1,<DDA                ABORT LOCATION
003262  0D5E  9F80 21A4                 STB     $B1,<RTRY               USE AS RETRY LOCATION
003263  0D60  9F80 218B                 STB     $B1,<NTST               ALSO USE AS NEXT TEST LOCATION
003264  0D62  9B80 10C2                 LAB     $B1,<RT4                RETRY ROUTINE (NOP)
003265  0D64  9F80 21A3                 STB     $B1,<RTRN
003266  0D66  D380 10B6                 LNJ     $B5,<SET                SET CNTRS, CLR FLAGS
003267  0D68  AC80 2185                 LDB     $B2,<KKPT               READ BUFFER POINTER
003268  0D6A  9C80 2184                 LDB     $B1,<JJPT               WRITE BUFFER POINTER
003269  0D6C  D380 123F                 LNJ     $B5,<ISB                CHECK DATA
003270  0D6E  D380 100F         DDA     LNJ     $B5,<SENSE              "NEXT ?;"
```

```
003271  0D70  0F80 0D0B              B       <MODEDI                 RETURN FOR NEXT DEBUG LINK.
003272                        ********************************************************************
003273                        * ERASE A BLOCK AND WAIT TILL DONE
003274                        *
003275  0D72  D380 0FA2       EE      LNJ     $B5,<LABEL              SET MAJOR ERROR LABEL
003276  0D74  4445            DE      DC      'DE'                    LABEL1
003277  0D75  D380 134D               LNJ     $B5,<ERAW               ERASE AND WAIT
003278  0D77  D380 100F               LNJ     $B5,<SENSE              "NEXT ?"
003279  0D79  D380 176A               LNJ     $B5,<EOTCHK             CHECK IF AT EOT
003280  0D7B  0F80 0D0B               B       <MODEDI                 RETURN
003281                        ********************************************************************
003282                        * FILL WRITE BUFFER WITH A RECORDS LENGTH OF FIXED DATA
003283                        *
003284  0D7D  8751            FFF     CL      =$R1
003285  0D7E  9C80 2184               LDB     $B1,<JJPT
003286  0D80  A800 2190               LDR     $R2,<PTRN               GET DATA
003287  0D82  A7DD            FFA     STH     $R2,$B1.+$R1            STORE THE DATA
003288  0D83  9900 219E               CMR     $R1,<RNG                DONE YET?
003289  0D85  027D                    BL      >FFA                    KEEP FILLING
003290  0D86  0F80 0D0B               B       <MODEDI                 RETURN
003291                        ********************************************************************
003292                        * FILL WRITE BUFFER WITH RECORDS LENGTH OF RANDOM DATA
003293                        *
003294  0D88  8751            GG      CL      =$R1
003295  0D89  9C80 2184               LDB     $B1,<JJPT
003296                        GGA     CALL    ZV$FR,TEMPA,DC1         GET A RAND NMBR
        0D8B  FBC0 0003
        0D8D  D380 0000         X
        0D8F  0F80
        0D90  21B7
        0D91  20EC
003297  0D92  A800 21B7               LDR     $R2,<TEMPA
003298  0D94  A7DD                    STH     $R2,$B1.+$R1            STORE IT IN BUFFER
003299  0D95  9900 219E               CMR     $R1,<RNG
003300  0D97  0200 0D8B               BL      <GGA
003301  0D99  0F80 0D0B               B       <MODEDI                 RETURN
003302                        ********************************************************************
003303                        * FILL WRITE BUFFER WITH RECORDS LENGTH OF ASCENDING COUNT
003304                        *
003305  0D9B  8751            II      CL      =$R1
003306  0D9C  9C80 2184               LDB     $B1,<JJPT
003307  0D9E  9791            IIA     STH     $R1,$B1.$R1
003308  0D9F  8AD1                    INC     =$R1
003309  0DA0  9900 219E               CMR     $R1,<RNG
003310  0DA2  0200 0D9E               BL      <IIA                    KEEP STUFFING
003311  0DA4  0F80 0D0B               B       <MODEDI                 RETURN
003312                        ********************************************************************
003313                        * ESTABLISH A WRITE BUFFER RANDOMLY LOCATED ABOVE ZV$LR
003314                        * IF 8K MACHINE, THIS HAS NO EFFECT.
003315                        *
003316  0DA6  FC80 0000         X     JJ      LDB     $B7,<ZV$HR              GET HMA IN B7
003317  0DA8  ABC0 2000         P             LAB     $B2,ZHCOMM+8192
003318  0DAA  FDD2                    CMB     $B7,=$B2
003319  0DAB  0380 0D0B               BLE     <MODEDI                 ONLY 8K, RETURN
003320                        *
003321  0DAD  9BC7 FB80               LAB     $B1,$B7.-1152           ROOM AT THE TOP FOR 2048+256 BYTES
003322  0DAF  9F80 218A               STB     $B1,<MXAD
003323  0DB1  D380 0FF7               LNJ     $B5,<GRNADR             GET RAND ADDRESS
003324  0DB3  9F80 2184               STB     $B1,<JJPT               STORE IT
003325  0DB5  9B80 1C66               LAB     $B1,<RBUF
003326  0DB7  9F80 2185               STB     $B1,<KKPT               RESTORE STANDARD READ BUFFER
003327  0DB9  D380 100F               LNJ     $B5,<SENSE              NEXT DESIRED ?
003328  0DBB  0F80 0D0B               B       <MODEDI                 RETURN
003329                        ********************************************************************
003330                        * ESTABLISH A RANDOMLY LOCATED READ BUFFER ABOVE ZV$LR
003331                        *
003332  0DBD  FC80 0000         X     KK      LDB     $B7,<ZV$HR              GET HMA IN B7
003333  0DBF  ABC0 2000         P             LAB     $B2,ZHCOMM+8192
003334  0DC1  FDD2                    CMB     $B7,=$B2
003335  0DC2  0380 0D0B               BLE     <MODEDI                 RETURN, ONLY 8K
003336                        *
003337  0DC4  9BC7 FB80               LAB     $B1,$B7.-1152           ROOM FOR 2048+256 BYTES
003338  0DC6  9F80 218A               STB     $B1,<MXAD
003339  0DC8  D380 0FF7               LNJ     $B5,<GRNADR             GET RANDOM ADDRESS
003340  0DCA  9F80 2185               STB     $B1,<KKPT               SAVE IT
003341  0DCC  9B80 17E6               LAB     $B1,<WBUF
003342  0DCE  9F80 2184               STB     $B1,<JJPT               RESTORE STANDARD READ BUFFER
003343  0DD0  D380 100F               LNJ     $B5,<SENSE              NEXT DESIRED ?
003344  0DD2  0F80 0D0B               B       <MODEDI                 RETURN
003345                        ********************************************************************
003346                        * SPARE SUBROUTINE
003347                        *
003348  0DD4  0F7F            LL      NOP     >$-1
003349  0DD5  0F7F                    NOP     >$-1
003350  0DD6  0F7F                    NOP     >$-1
003351  0DD7  0F7F                    NOP     >$-1
003352  0DD8  0F7F                    NOP     >$-1
003353  0DD9  0F7F                    NOP     >$-1
003354  0DDA  0F7F                    NOP     >$-1
003355  0DDB  0F7F                    NOP     >$-1
003356  0DDC  0F7F                    NOP     >$-1
003357  0DDD  0F7F                    NOP     >$-1
003358  0DDE  0F7F                    NOP     >$-1
003359  0DDF  0F7F                    NOP     >$-1
003360  0DE0  0F7F                    NOP     >$-1
003361  0DE1  0F7F                    NOP     >$-1
003362  0DE2  0F7F                    NOP     >$-1
003363  0DE3  0F7F                    NOP     >$-1
003364  0DE4  0F7F                    NOP     >$-1
003365  0DE5  0F7F                    NOP     >$-1
003366  0DE6  0F7F                    NOP     >$-1
003367  0DE7  0F7F                    NOP     >$-1
003368  0DE8  0F7F                    NOP     >$-1
003369  0DE9  0F7F                    NOP     >$-1
003370  0DEA  0F7F                    NOP     >$-1
003371  0DEB  0F7F                    NOP     >$-1
003372  0DEC  0F7F                    NOP     >$-1
003373  0DED  0F7F                    NOP     >$-1
003374  0DEE  0F7F                    NOP     >$-1
003375  0DEF  0F7F                    NOP     >$-1
003376  0DF0  0F7F                    NOP     >$-1
003377  0DF1  0F7F                    NOP     >$-1
003378  0DF2  0F7F                    NOP     >$-1
```

```
003379  0DF3  0F7F                    NOP     >$-1
003380  0DF4  0F7F                    NOP     >$-1
003381  0DF5  0F7F                    NOP     >$-1
003382  0DF6  0F7F                    NOP     >$-1
003383  0DF7  0F7F                    NOP     >$-1
003384  0DF8  0F7F                    NOP     >$-1
003385  0DF9  D380 100F               LNJ     $B5,<SENSE              "NEXT ?"
003386  0DFB  0F81 FF0F               B       MODEDI                  RETURN
003387                          *
003388                          * ROOM FOR CONSTANTS, ETC.
003389                          *
003390  0DFD  0000                    RESV    1,0
003391  0DFE  0000                    RESV    1,0
003392  0DFF  0000                    RESV    1,0
003393  0E00  0000                    RESV    1,0
003394  0E01  0000                    RESV    1,0
003395                          ********************************************************************
003396                          * WRITE A FILE MARK
003397                          *
003398  0E02  D380 0FA2         MM    LNJ     $B5,<LABEL              MAJOR ERROR LABEL
003399  0E04  444D              DM    DC      'DM'                    LABEL1
003400  0E05  D380 131A               LNJ     $B5,<WFM                WRITE FILE MARK
003401  0E07  D380 0FEC               LNJ     $B5,<INSTAT             GET STATUS
003402  0E09  D380 100F               LNJ     $B5,<SENSE              "NEXT ?"
003403  0E0B  D380 176A               LNJ     $B5,<EOTCHK             CHECK IF AT EOT
003404  0E0D  0F80 0D0B               B       <MODEDI                 RETURN
003405                          ********************************************************************
003406                          * GENERATE A SET OF RANDOM PARAMETERS (RNG, PTRN)
003407                          *
003408  0E0F  1C01              NN    LDV     $R1,=1
003409  0E10  9F00 21B7               STR     $R1,<TEMPA              RANGE = 1
003410                                CALL    ZV$FR,TEMPB,TEMPA       GET RANDOM NMBR IN TEMPB
        0E12  FBC0 0003
        0E14  D380 0000         X
        0E16  0F80
        0E17  21B6
        0E18  21B7
003411                          *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
003412                          * RANGE
003413                          *
003414  0E19  9800 21B8               LDR     $R1,<TEMPB              GET RAND DATA
003415  0E1B  9570 07FF               AND     $R1,=Z'07FF'            LIMIT RANGE TO 12-800 HEX
003416  0E1D  1E01                    ADV     $R1,=1
003417  0E1E  1D11                    CMV     $R1,=17                 =11 HEX
003418  0E1F  0A02                    BAG     >NNA                    B IF ALREADY WITHIN RANGE
003419  0E20  1E11                    ADV     $R1,=17                 PUT WITHIN RANGE
003420  0E21  9F00 219E        NNA    STR     $R1,<RNG                SAVE RANDOM RANGE
003421                          *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
003422                          * PATTERN
003423                          *
003424  0E23  9280 21B8               LLH     $R1,<TEMPB              GET LEFT BYTE OF RAND NMBR
003425  0E25  9F00 2190               STR     $R1,<PTRN               STORE AS PATTERN
003426  0E27  D380 100F               LNJ     $B5,<SENSE              NEXT?
003427  0E29  0F80 0D0B               B       <MODEDI                 RETURN
003428                          ********************************************************************
003429                          * BACKSPACE A FILE
003430                          *
003431  0E2B  D380 0FA2         OO    LNJ     $B5,<LABEL
003432  0E2D  444F              DO    DC      'DO'                    LABEL1
003433  0E2E  D380 13F5               LNJ     $B5,<BSFW               BACKSPACE AND WAIT TILL DONE
003434  0E30  D380 100F               LNJ     $B5,<SENSE              NEXT?
003435  0E32  0F80 0D0B               B       <MODEDI                 RETURN
003436                          ********************************************************************
003437                          * PRINT PARAMETER LIST (DRIVE, RNG, PTRN, JJPT, KKPT) ALL HEX
003438                          *
003439                          PP      CALL    ZV$T,ZV$TC,MSPARM       PRINT "PARMS"
        0E34  FBC0 0003
        0E36  D380 0000         X
        0E38  0F80
        0E39  2271
003440                          *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
003441                          * DRIVE
003442                          *
003443  0E3A  9800 20F5               LDR     $R1,<DRIVE
003444  0E3C  1047                    SUR     $R1,7                   RIGHT JUSTIFY
003445  0E3D  9F00 21B7               STR     $R1,<TEMPA
003446                                CALL    ZV$HA,TEMPA,VRBL        CONVERT TO ASCII HEX
        0E3F  FBC0 0003
        0E41  D380 0000         X
        0E43  0F80
        0E44  21B7
        0E45  21BB
003447                                CALL    ZV$T,VRBL+2             PRINT DRIVE NMBR
        0E46  FBC0 0003
        0E48  D380 0000         X
        0E4A  0F80
        0E4B  21BB
003448                          *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
003449                          * RANGE, BYTES (HEX)
003450                          *
003451  0E4C  D380 145F               LNJ     $B5,<BLANKS             PRINT 2 BLANKS
003452                                CALL    ZV$HA,RNG,VRBL          CONVERT TO ASCII HEX
        0E4E  FBC0 0003
        0E50  D380 0000         X
        0E52  0F80
        0E53  219E
        0E54  21BB
003453                                CALL    ZV$T,VRBL+1             PRINT RANGE, HEX BYTES (800 MAX)
        0E55  FBC0 0003
        0E57  D380 0000         X
        0E59  0F80
        0E5A  21BC
003454                          *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
003455                          * PATTERN
003456                          *
003457  0E5B  D380 145F               LNJ     $B5,<BLANKS             PRINT 2 BLANKS
003458                                CALL    ZV$HA,ZV$HZ,PTRN,VRBL   CONVERT TO ASCII HEX
        0E5D  FBC0 0003
        0E5F  D380 0000         X
        0E61  0F80
        0E62  2190
        0E63  21BB
003459                                CALL    ZV$T,VRBL+2             PRINT PATTERN, HEX
```

```
                0E64  FBC0 0003
                0E66  D380 0000       X
                0E68  0F80
                0E69  21BD
003460                                *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
003461                                * WRITE BUFFER (JJPT)
003462                                *
003463  0E6A  D380 145F                       LNJ   $B5,<BLANKS          PRINT 2 BLANKS
003466                                PPA     NULL
003467                                        CALL  ZV$IH,ZV$THZ,JJPT+$AF-1,DCB5
                0E6C  FBC0 0003
                0E6E  D380 0000       X
                0E70  0F80
                0E71  21B4
                0E72  20F2
003468                                *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
003469                                * READ BUFFER (KKPT)
003470                                *
003471  0E73  D380 145F                       LNJ   $B5,<BLANKS          PRINT 2 BLANKS
003474                                PPB     NULL
003475                                        CALL  ZV$IH,ZV$THZ,KKPT+$AF-1,DCB5
                0E75  FBC0 0003
                0E77  D380 0000       X
                0E79  0F80
                0E7A  2185
                0E7B  20F2
003476                                *
003477  0E7C  D380 100F                       LNJ   $B5,<SENSE           NEXT?
003478  0E7E  0F81 FE8C                       B     MODEDI               RETURN
003479                                ********************************************************************
003480                                * TRANSFER CONTENTS OF READ BUFFER TO WRITE BUFFER
003481                                *
003482  0E80  9800 219E               QQ      LDR   $R1,<RNG             GET RANGE
003483  0E82  1F04                           MLV   $R1,=4               MODIFY FOR CALL STATEMENT
003484  0E83  9F00 21B7                      STR   $R1,<TEMPA
003485                                        CALL  ZV$MB,TEMPA,KKPT,JJPT MOVE BYTES
                0E85  FBC0 0003
                0E87  D380 0000       X
                0E89  0F80
                0E8A  21B7
                0E8B  2185
                0E8C  2184
003486  0E8D  D380 100F                       LNJ   $B5,<SENSE           NEXT?
003487  0E8F  0F81 FE7B                       B     MODEDI               RETURN
003488                                ********************************************************************
003489                                * READ A RECORD INTO DEBUG READ BUFFER (*KKPT)
003490                                *
003491  0E91  D380 0FA2               RR      LNJ   $B5,<LABEL
003492  0E93  4452                    DR      DC    'DR'                 LABEL1
003493  0E94  9C80 2185                      LDB   $B1,<KKPT            READ BUFFER
003494  0E96  9F80 21B3                      STB   $B1,<SLDB            CURRENT DATA BUFFER
003495  0E98  9F80 0EA7                      STB   $B1,<RRA+4+$AF       BUF ADR FOR ZV$F
003496  0E9A  9B80 0EBE                      LAB   $B1,<RRD
003497  0E9C  9F80 21A4                      STB   $B1,<RTRY            RETRY LOC (NO RETRY)
003498  0E9E  9B80 10C2                      LAB   $B1,<RT4             NOP
003499  0EA0  9F80 21A3                      STB   $B1,<RTRN            RETRY ROUTINE (NOP)
003500                                *
003501                                RRA     CALL  ZV$F,RBUF,X3333,LENGTH (BUF, DATA, RANGE-WORDS)
                0EA2  FBC0 0003
                0EA4  D380 0000       X
                0EA6  0F80
                0EA7  1C66
                0EA8  21BF
                0EA9  2189
003502                                *
003503  0EAA  D380 10B6                       LNJ   $B5,<SET
003504  0EAC  8AB0 210E                       INC   <DXRN,$R3            BUMP RECORD COUNTER
003505  0EAE  8188 2185               RRB     IOLD  **<KKPT,<IOREAD,<RNG
                0EB0  0000 22BB
                0EB2  0000 219E
003506  0EB4  07FA                            BIOF  >RRB                 KEEP TRYING
003507  0EB5  8000 229F               RRC     IO    <READFD,<OTTASK      START IO READ
                0EB7  0000 22C2
003508  0EB9  07FC                            BIOF  >RRC                 KEEP TRYING
003509  0EBA  D380 0FEC                       LNJ   $B5,<INSTAT          WAIT FOR BUSY
003510                                *
003511  0EBC  D380 1480                       LNJ   $B5,<ERCK            CHECK FOR ERRORS
003512  0EBE  D380 100F               RRD     LNJ   $B5,<SENSE           NEXT ?
003513  0EC0  D380 176A                       LNJ   $B5,<EOTCHK          CHECK IF AT EOT
003514  0EC2  0F80 0D0B                       B     <MODEDI              RETURN
003515                                ********************************************************************
003516                                * INPUT BOTH STATUS WORDS (MAY MODIFY PREVIOUSLY RECEIVED STATUS)
003517                                *
003518  0EC4  D380 0FEC               SS      LNJ   $B5,<INSTAT          GET STATUS WORDS
003519  0EC6  D380 100F                       LNJ   $B5,<SENSE           NEXT ?
003520  0EC8  0F80 0D0B                       B     <MODEDI              RETURN
003521                                ********************************************************************
003522                                * BACKSPACE A RECORD
003523                                *
003524  0ECA  D380 0FA2               TT      LNJ   $B5,<LABEL
003525  0ECC  4454                    DT      DC    'DT'                 LABEL1
003526  0ECD  D380 0FEC                       LNJ   $B5,<INSTAT          GET STATUS
003527  0ECF  8280 21B4                       LB    <STAT1,=Z'0200'
                0ED1  0200
003528  0ED2  0503                            BBT   >TTA                 B IF B-O-T
003529  0ED3  D380 142B                       LNJ   $B5,<BSRW            BACKSPACE AND WAIT
003530  0ED5  D380 100F               TTA     LNJ   $B5,<SENSE           NEXT ?
003531  0ED7  0F80 0D0B                       B     <MODEDI              RETURN
003532                                ********************************************************************
003533                                * REWIND TO B-O-T
003534                                *
003535  0ED9  D380 0FA2               UU      LNJ   $B5,<LABEL
003536  0EDB  4455                    DU      DC    'DU'                 LABEL1
003537  0EDC  8000 2299                       IO    <REWIND,<OTTASK
                0EDE  0000 22C2
003538  0EE0  0705                            BIOT  >UUA                 B IF OK
003539  0EE1  D380 1030                       LNJ   $B5,<FNER            SHOULD HAVE BEEN RDY FOR REWIND
003540  0EE3  3033                    FNER03  DC    '03'                 LABEL2
003541  0EE4  0F8F                            B     >UUC                 ABORT
003542  0EE5  D380 0FEC               UUA     LNJ   $B5,<INSTAT          GET STATUS
003543  0EE7  9800 21B4                       LDR   $R1,<STAT1
003544  0EE9  9970 8200                       CMR   $R1,=Z'8200'         RDY, B-O-T
003545  0EEB  0904                            BE    >UUB                 B IF OK
```

```
003546  0EEC  D380 1030          LNJ    $B5,<FNER          STAT 1 NG AFTER REWIND
003547  0EEE  3038               DC     '08'               LABEL2
003548  0EEF  8730 20FE  UUB     CL     <DXFN.$R3          RESET FILE COUNTER
003549  0EF1  8730 210E          CL     <DXRN.$R3          RESET RECORD COUNTER
003550  0EF3  D380 100F  UUC     LNJ    $B5,<SENSE         NEXT ?
003551  0EF5  0F80 0D0B          B      <MODEDI            RETURN
003552                  ********************************************************************
003553                  * FORWARD SPACE A RECORD
003554                  *
003555  0EF7  D380 0FA2  VV      LNJ    $B5,<LABEL
003556  0EF9  4456       DV      DC     'DV'               LABEL1
003557  0EFA  D380 13C1          LNJ    $B5,<FSRW          FORWARD SPACE AND WAIT
003558  0EFC  D380 100F          LNJ    $B5,<SENSE         NEXT ?
003559  0EFE  D380 176A          LNJ    $B5,<EOTCHK        CHECK IF AT EOT
003560  0F00  0F80 0D0B          B      <MODEDI            RETURN
003561                  ********************************************************************
003562                  * WRITE A RECORD FROM DEBUG WRITE BUFFER (*JJPT)
003563                  *
003564  0F02  D380 0FA2  WW      LNJ    $B5,<LABEL
003565  0F04  4457       DW      DC     'DW'               LABEL1
003566  0F05  9B80 0F25          LAB    $B1,<WWC           RETRY LOCATION
003567  0F07  9F80 21A4          STB    $B1,<RTRY
003568  0F09  9F80 218B          STB    $B1,<NTST          IS ALSO ABORT LOCATION
003569  0F0B  9B80 10C2          LAB    $B1,<RT4           NOP ROUTINE
003570  0F0D  9F80 21A3          STB    $B1,<RTRN          RETRY ROUTINE IS A NOP
003571  0F0F  9F80 20E6          STB    $B1,<ABRN          SAME FOR ABORT ROUTINE
003572                  *
003573  0F11  D380 1469          LNJ    $B5,<ANB           CHECK FOR ACT, N/BSY
003574  0F13  8AB0 210E          INC    <DXRN.$R3          BUMP RECORD COUNTER
003575  0F15  8188 2184  WWA     IOLD   **JJPT,<IOWRIT,<RNG
        0F17  0000 22BC
        0F19  0000 219E
003576  0F1B  07FA               BIOF   >WWA               KEEP TRYING
003577  0F1C  8000 22A2  WWB     IO     <WRITE,<OUTTASK    START IO WRITE
        0F1E  0000 22C2
003578  0F20  07FC               BIOF   >WWB               KEEP TRYING
003579  0F21  D380 0FEC          LNJ    $B5,<INSTAT        WAIT FOR N/BSY, GET STATUS
003580  0F23  D380 1480          LNJ    $B5,<ERCK          CHECK FOR ERRORS
003581  0F25  D380 100F  WWC     LNJ    $B5,<SENSE         NEXT ?
003582  0F27  D380 176A          LNJ    $B5,<EOTCHK        CHECK IF AT EOT
003583  0F29  0F80 0D0B          B      <MODEDI            RETURN
003584                  ********************************************************************
003585                  * EXIT TO "NEXT ?:"
003586                  *
003587  0F2B  0F80 0229  XX      B      <RESTRT            RETURN TO "NEXT ?:" (VIA RESTART)
003588                  ********************************************************************
003589                  * FORWARD SPACE A FILE
003590                  *
003591  0F2D  D380 0FA2  YY      LNJ    $B5,<LABEL
003592  0F2F  4459       DY      DC     'DY'               LABEL1
003593  0F30  D380 1375          LNJ    $B5,<FSFW          FORWARD SPACE AND WAIT
003594  0F32  D380 100F          LNJ    $B5,<SENSE         NEXT ?
003595  0F34  D380 176A          LNJ    $B5,<EOTCHK        CHECK IF AT EOT
003596  0F36  0F80 0D0B          B      <MODEDI            RETURN
003597                  ********************************************************************
003598                  * PRINT UNIT, FILE, RECORD, REMAINING RANGE, STATUS 1, STATUS 2.
003599                  *
003600  0F38  D380 136E  ZZ      LNJ    $B5,<GETRNG        GET REMAINING RANGE, STORE IN "RANGE"
003601                           CALL   ZV$I.ZV$IC,MSSTAT  PRINT "  STAT :"
        0F3A  FBC0 0003
        0F3C  D380 0000     X
        0F3E  0F80
        0F3F  2294
003602                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
003603                  * UNIT
003604                  *
003605  0F40  8F00 21B7          STR    $R3,<TEMPA         SAVE UNIT NMBR (INDEX)
003606                           CALL   ZV$IH.ZV$TD,TEMPA  UNIT
        0F42  FBC0 0003
        0F44  D380 0000     X
        0F46  0F80
        0F47  21B7
003607                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
003608                  * FILE NUMBER (HEX)
003609                  *
003610  0F48  E830 20FE          LDR    $R6,<DXFN.$R3      GET FILE COUNT
003611  0F4A  F830 210E          LDR    $R7,<DXRN.$R3      GET RECORD COUNT
003612  0F4C  8D00 21B7          SDI    <TEMPA             STORE IN TEMPA AND TEMPB
003613  0F4E  D380 145F          LNJ    $B5,<BLANKS
003614                           CALL   ZV$IH,TEMPA        FILE NUMBER
        0F50  FBC0 0003
        0F52  D380 0000     X
        0F54  0F80
        0F55  21B7
003615                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
003616                  * RECORD NUMBER (HEX)
003617                  *
003618  0F56  D380 145F          LNJ    $B5,<BLANKS
003619                           CALL   ZV$IH,TEMPB        RECORD NUMBER
        0F58  FBC0 0003
        0F5A  D380 0000     X
        0F5C  0F80
        0F5D  21B8
003620                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
003621                  * RANGE REMAINING FROM PREVIOUS OPERATION (HEX)
003622                  *
003623  0F5E  D380 145F          LNJ    $B5,<BLANKS
003624                           CALL   ZV$IH,RANGE        RANGE REMAINING
        0F60  FBC0 0003
        0F62  D380 0000     X
        0F64  0F80
        0F65  219D
003625                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
003626                  * STAT 1
003627                  *
003628  0F66  D380 145F          LNJ    $B5,<BLANKS
003629                           CALL   ZV$IH.ZV$THZ,STAT1 STAT1
        0F68  FBC0 0003
        0F6A  D380 0000     X
        0F6C  0F80
        0F6D  21B4
003630                  *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
003631                  * STAT 2
```

```
003632                                  *
003633  0F6E  D380 145F                         LNJ     $B5,<BLANKS
003634                                          CALL    ZV$TH.ZV$THZ,STAT2      STAT2
        0F70  FBC0 0003
        0F72  D380 0000         A
        0F74  0F80
        0F75  21B5
003635                                  *
003636  0F76  D380 100F                         LNJ     $B5,<SENSE              NEXT ?
003637  0F78  0F80 0D0B                         B       <MODEDI                 RETURN
003638                                  ********************************************************************************
```

```
003639                                  /*****************************************************************
003640                                  * INPUT A HEX FILE LIMIT FOR USE IN MODES A Q R AND W
003641                                  *
003642  0F7A  0380 1626                 MODEF   LNJ     $B5,<CRLF               DO A LINE FEED
003643  0F7C  8980 217F                         CMZ     <FILES
003644  0F7E  0908                              BE      >MODEFA                B IF E-O-T
003645                                          CALL    ZV$IH,FILES            REPORT CURRENT HEX FILE LIMIT
        0F7F  FBC0 0003
        0F81  0380 0000         X
        0F83  0F80
        0F84  217F
003646  0F85  0F87                              B       >MODEFB
003647                                  MODEFA  CALL    ZV$I,MSEOT             PRINT "E-O-T"
        0F86  FBC0 0003
        0F88  0380 0000         X
        0F8A  0F80
        0F8D  2235
003648                                  MODEFB  CALL    ZV$I,ZV$Q,MSFILE       ASK "FILES ?:"
        0F8C  FBC0 0003
        0F8E  0380 0000         X
        0F90  0F80
        0F91  223F
003649                                          CALL    ZV$IH,FILES            INPUT FILE LIMIT (HEX)
        0F92  FBC0 0003
        0F94  0380 0000         X
        0F96  0F80
        0F97  217F
003650  0F98  0F80 0237                        B       <NEXT                  GO BACK TO "NEXT ?:"
003651                                  *****************************************************************
```

```
003652                                  /*****************************************************************
003653                                  *
003654                                  * SUBROUTINES
003655                                  *
003656                                  *****************************************************************
003657                                  * NON-EXISTANT RESOURCE, TRAP 15 HANDLER
003658                                  *
003659  0F9A  9B80 0100                 TH15    LAB     $B1,<START              RESTART LOCATION
003660  0F9C  9FC3 0001                         STB     $B1,$B3.$AF             SAVE IN CURRENT TSAP
003661  0F9E  D380 1030                         LNJ     $B5,<FNER               MISSING RESOURCE TRAP
003662  0FA0  3636                     FNER66  DC      '66'                    LABEL2
003663  0FA1  0003                             RTT                             RETURN FROM TRAP, GO TO "NEXT ?:"
003664                                  *****************************************************************
003665                                  * SET UP ERROR REPORTING LABEL
003666                                  *
003667  0FA2  D875                      LABEL   LDR     $R5,+$B5
003668  0FA3  0F00 2186                         STR     $R5,<LABEL1
003669  0FA5  8385                             JMP     $B5                     RETURN
003670                                  *****************************************************************
003671                                  * HANDLE RTC RUPT AND CALIBRATE CLOCK (AT LEVEL 5)
003672                                  *
003673  0FA6  0005                      RTCFC   RTCF                            STOP THE CLOCK
003674  0FA7  C370 FC18                         DIV     $R4,=-1000
003675  0FA9  CF00 0FC4                         STR     $R4,<SYNCHC             TIME FOR 1 MILLISECOND
003676  0FAB  9B80 0142                         LAB     $B1,<RESUME
003677  0FAD  9F80 230F                         STB     $B1,<SA15P              PREPARE FOR SUSPEND
003678  0FAF  8E70 800F                 LEVXG   LEV     =Z'8000'+15             SUSPEND TO LEVEL 15
003679  0FB1  0000                             HLT                             DIDN'T SET UP FOR SECOND INTERRUPT
003680  0FB2  0FFF                             B       >$-1
003681                                  *****************************************************************
003682                                  * CALIBRATE THE CPU FOR ONE SECOND
003683                                  *
003684  0FB3  6754                      SYNCH   CL      =$R4                    LOOPS FOR 1 SEC
003685  0FB4  0004                      RTCNB   RTCN                            START THE CLOCK
003686  0FB5  9900 0000         X      SYNCHA  CMR     $R1,<ZHRTCC             SYNCHRONIZE THE CLOCK TICKS
003687  0FB7  0900 0FB5                         BE      <SYNCHA                 WAIT FOR FIRST TICK
003688  0FB9  1CFF                              LDV     $R1,=-1                 IN CASE RTC DOESN'T RUPT
003689                                  *
003690  0FBA  2CE7                      SYNCHB  LDV     $R2,=-25                -25
003691  0FBB  2780 0FBB                         BINC    $R2,<$                  LONG NOP
003692  0FBD  4780 0FBA                         BINC    $R4,<SYNCHB             COUNT LOOPS/SEC (/MS)
003693                                  *
003694                                  * IF DOING INITIAL CALIBRATION, SHOULD RUPT OUT OF LOOP TIMEC.
003695                                  *
003696  0FBF  C800 0FC4                         LDR     $R4,<SYNCHC             NMBR OF LOOPS/MS
003697  0FC1  1780 0FBA                         BINC    $R1,<SYNCHB             NMBR OF MSECS
003698  0FC3  8385                             JMP     $B5                     RETURN
003699                                  *
003700  0FC4  0000                      SYNCHC  RESV    1,0                     NEG NMBR TO RE-INIT FOR 1 MSEC
003701                                  *****************************************************************
003702                                  * TIMEOUT FOR N MILLISECONDS
003703                                  *
003704  0FC5  9875                      TIME    LDR     $R1,+$B5                GET NMBR OF MSECS
003705  0FC6  8251                             NEG     =$R1
003706  0FC7  C800 0FC4                         LDR     $R4,<SYNCHC             SET UP FOR FIRST MSEC
003707  0FC9  0F80 0FBA                         B       <SYNCHB                 TIMEOUT
003708                                  *****************************************************************
003709                                  * TYPE OUT ASCII CONTENTS OF R1
003710                                  *
003711  0FCB  DF80 0FDA                 VDTR    STB     $B5,<VDTRB5
003712  0FCD  9F00 0FD8                         STR     $R1,<VDTRA
003713                                          CALL    ZV$T,VDTRA
       0FCF  FBC0 0003
       0FD1  D380 0000          X
       0FD3  0F80
       0FD4  0FD8
003714  0FD5  DC80 0FDA                         LDB     $B5,<VDTRB5
003715  0FD7  8385                             JMP     $B5                     RETURN
003716                                  *
003717  0FD8  0000                      VDTRA   RESV    1,0
003718  0FD9  2424                             DC      '$$'
003719  0FDA  0000                      VDTRB5  RESV    $AF,0
003720                                  *****************************************************************
003721                                  * DETERMINE IF MODE R, W, A OR Q
003722                                  *
003723                                  *       LNJ   $B5,<RWAQ
003724                                  *       "R" RETURN
003725                                  *       "W" RETURN
003726                                  *       "A" RETURN
003727                                  *       "Q" RETURN
003728                                  *
003729  0FDB  9830 2102                 RWAQ    LDR     $R1,<DXMD.$R3           GET MODE
003730  0FDD  8752                             CL      =$R2                    INDEX FOR RETURN
003731  0FDE  1048                             SOR     $R1,8                   RIGHT JUSTIFY
003732  0FDF  9970 0052                         CMR     $R1,=X'52'              ASCII "R"
003733  0FE1  090A                              BE      >RWAQZ
003734  0FE2  8AD2                              INC     =$R2
003735  0FE3  9970 0057                         CMR     $R1,=X'57'              ASCII "W"
003736  0FE5  0906                              BE      >RWAQZ
003737  0FE6  8AD2                              INC     =$R2
003738  0FE7  9970 0041                         CMR     $R1,=X'41'              ASCII "A"
003739  0FE9  0902                              BE      >RWAQZ
003740  0FEA  8AD2                              INC     =$R2                    MUST BE "Q"
003741  0FEB  83A5                      RWAQZ   JMP     $B5,$R2                 RETURN
003742                                  *****************************************************************
003743                                  * GET TWO STATUS WORDS
003744                                  *
003745  0FEC  8000 21B5                 INSTAT  IO      <STAT2,<INSTW2          GET STATUS WORD 2
       0FEE  0000 22CD
003746  0FF0  07FC                              BIOF    >INSTAT
003747  0FF1  8000 21B4                 ISTAT1  IO      <STAT1,<INSTW1          GET STATUS WORD 1
       0FF3  0000 22CC
003748  0FF5  07FC                              BIOF    >ISTAT1
003749  0FF6  8385                              JMP     $B5                     RETURN
003750                                  *****************************************************************
003751                                  * GENERATE RANDOM ADDRESS ABOVE ZV$LR BUT LESS THAN MXAD
003752                                  * (IE. LEAVE ROOM FOR 2048+256 BYTES)
003753                                  * RETURN WITH RANDOM ADDRESS IN B1.
003754                                  *
003755  0FF7  DF80 100E                 GRNADR  STB     $B5,<GRNB5
003756  0FF9  9C80 2194                         LDB     $B1,<RANADR             GET OLD RANDOM ADDRESS
003757  0FFB  9BC1 0737                 GRNA    LAB     $B1,$B1.1847            ADD A PRIME NUMBER
003758  0FFD  9D80 0000          X              CMB     $B1,<ZV$LR
```

```
003759  0FFF  027C                    BL      >GRNA                   B IF TOO SMALL
003760  1000  9D80 218A                CMB     $B1,<MXAD
003761  1002  0207                    BL      >GRNC                   B IF OK
003762                          *
003763  1003  9DC1 F000        GRNB    LAB     $B1,$B1,-4096           START OVER
003764  1005  9D80 0000      X         CMB     $B1,<ZV$LR
003765  1007  027C                    BG      >GRNB                   B IF STILL TOO LARGE
003766  1008  0FF3                    B       >GRNA                   START AGAIN
003767                          *
003768  1009  9F80 2194        GRNC    STB     $B1,<RANADR             SAVE NEW RANDOM ADDRESS
003769  100B  DC80 100E                LDB     $B5,<GRNB5
003770  100D  8385                    JMP     $B5                     RETURN WITH NUMBER IN B1
003771                          *
003772  100E  0000              GRNB5   RESV    $AF,0
003773                          ********************************************************************
003774                          * CHECK IF OPERATOR WISHES TO INTERVENE
003775                          *
003776  100F  DF80 102F        SENSE   STB     $B5,<SENSB5             SAVE B5 FOR RETURN
003777  1011  BF00 2193                STR     $R3,<R3HOLD             SAVE "UNIT" INDEX FOR RETURN FROM "NEXT"
003778                                  CALL    ZV$BRK                  CHECK THE CONSOLE
        1013  FBF0 0001
        1015  D380 0000      X
003779  1017  DC80 102F                LDB     $B5,<SENSB5             RETURN ADDRESS FOR IN-LINE RETURN
003780  1019  8980 0000      X         CMZ     <ZV$BKF                 CHECK THE CONSOLE
003781  101B  0913                    BE      >SENSB                  B IF NO BREAK, RETURN IN-LINE
003782                          *
003783  101C  9080 2186                LDH     $R1,<LABEL1
003784  101E  1D44                    CMV     $R1,='D'                ASCII "D"
003785  101F  090D                    BE      >SENSA                  MODE "D", RESTART
003786                          *
003787  1020  DB80 0237                LAB     $B5,<NEXT
003788  1022  8800 218F                LBF     <PFLAG,=1               CHECK IF MODE "P" AND CLR FLAG
        1024  0001
003789  1025  0589                    BBF     >SENSB                  B TO NEXT IF NOT "P"
003790                          *
003791  1026  8900 218F                LBT     <PFLAG,=2               SET BRK-DETECTED FLAG
        1028  0002
003792  1029  DC80 102F                LDB     $B5,<SENSB5
003793  102B  0F83                    B       >SENSB                  RETURN IN-LINE. BRK LATER ON
003794                          *
003795  102C  DB80 0229        SENSA   LAB     $B5,<RESTRT             MODE "D", RESTART
003796  102E  8385              SENSB   JMP     $B5                     RETURN SOMEWHERE.
003797                          *
003798  102F  0000              SENSB5  RESV    $AF,0                   HOLD B5 HERE
003799                          ********************************************************************
003800                          * ERROR REPORTING SUBROUTINE
003801                          *
003802  1030  9875              FNER    LDR     $R1,+$B5                GET MINOR LABEL
003803  1031  9F00 2187                STR     $R1,<LABEL2
003804  1033  DF80 10AA                STB     $B5,<FNERB5
003805  1035  ABC5 FFFD                LAB     $B2,$B5,-2-$AF          ADDRESS OF "LNJ FNER"
003806  1037  AF80 107C                STB     $B2,<FNERF              SAVE FOR REPORT
003807  1039  8A80 2113        RPRT    INC     <ERCT                   TALLY THE ERROR
003808  103B  BF00 2114                STR     $R3,<ERDRIV             SAVE DRIVE NMBR, HEX
003809  103D  9830 20FE                LDR     $R1,<DXFN.$R3
003810  103F  9F00 2115                STR     $R1,<ERFILE             SAVE FILE NMBR, HEX
003811  1041  9830 210E                LDR     $R1,<DXRN.$R3
003812  1043  9F00 2117                STR     $R1,<ERREC              SAVE RECORD NMBR, HEX
003813  1045  9800 2113                LDR     $R1,<ERCT
003814  1047  1D0A                    CMV     $R1,=10
003815  1048  0A00 1068                BAG     <FNERA                  B IF TABLE FULL
003816                          *
003817                          * SAVE ERROR PARAMETERS FOR MODE "P"
003818                          *
003819  104A  9C80 217D                LDB     $B1,<ERTBPT
003820                          *
003821  104C  9800 2186                LDR     $R1,<LABEL1
003822  104E  9F71                    STR     $R1,+$B1                LABEL1, ASCII
003823                          *
003824  104F  9800 2187                LDR     $R1,<LABEL2
003825  1051  9F71                    STR     $R1,+$B1                LABEL2, ASCII
003826                          *
003827  1052  AFF1                    STB     $B2,+$B1                LOCATION, HEX
003828                          *
003829  1053  BF71                    STR     $R3,+$B1                DRIVE NUMBER, HEX
003830                          *
003831  1054  9830 20FE                LDR     $R1,<DXFN.$R3
003832  1056  9F71                    STR     $R1,+$B1                FILE NUMBER, HEX
003833                          *
003834  1057  9830 210E                LDR     $R1,<DXRN.$R3
003835  1059  9F71                    STR     $R1,+$B1                RECORD NUMBER, HEX
003836                          *
003837  105A  BC80 21B3                LDB     $B3,<SLDB               DATA BUFFER, HEX
003838  105C  BFF1                    STB     $B3,+$B1
003839                          *
003840  105D  9800 21B4                LDR     $R1,<STAT1
003841  105F  9F71                    STR     $R1,+$B1                STATUS WORD 1, HEX
003842                          *
003843  1060  9800 21B5                LDR     $R1,<STAT2
003844  1062  9F71                    STR     $R1,+$B1                STATUS WORD 2, HEX
003845                          *
003846  1063  9830 2106                LDR     $R1,<DXPS.$R3
003847  1065  9F71                    STR     $R1,+$B1                PASS COUNT, DECIMAL
003848                          *
003849  1066  9F80 217D                STB     $B1,<ERTBPT             SAVE POINTER FOR NEXT ERROR
003850                          *
003851                          * REPORT ERROR UNLESS SUPPRESSED
003852                          *
003853  1068  8280 21B6        FNERA   LB      <SUPRES,=1
        106A  0001
003854  106D  0500 10A5                BBT     <FNERC                  B IF SUPPRESSED, IF.., RETURN
003855                          *
003856                          * THIS IS ALSO THE ENTRY POINT FOR MODE "P"
003857                          *
003858  106D  DF80 10AA        FNERMP  STB     $B5,<FNERB5             HOLD B5 FOR RETURN
003859  106F  8900 2118                LBT     <ERRFLG,=1              CHECK "FIRST ERROR" FLAG
        1071  0001
003860  1072  0507                    BBT     >FNERB                  B IF AN ERROR ALREADY REPORTED
003861                                  CALL    ZV$T,ZV$TC,MSHEAD       (ERROR HEADING)
        1073  FBC0 0003
        1075  D380 0000      X
        1077  0F80
        1078  21FA
```

```
003862  1079  D380 0000      X     FNERB  LNJ    $B5,<ZV$ER          REPORT ERROR LABEL AND LOCATION
003863  107B  0F83                  B      >FNERE
003864  107C  0001                  FNERF  RESV   $AF,1               ERROR LOC
003865  107D  2186                  DC     <LABEL1                  LABEL
003866                               FNERE  CALL   ZV$TH,ZV$TD,ERDRIV  REPORT DRIVE (UNIT)
        107E  FBC0 0003
        1080  D380 0000      X
        1082  0F80
        1083  2114
003867                               CALL   ZV$TH,ERFILE              REPORT FILE
        1084  FBC0 0003
        1086  D380 0000      X
        1088  0F80
        1089  2115
003868                               CALL   ZV$TH,ERREC               REPORT RECORD
        108A  FBC0 0003
        108C  D380 0000      X
        108E  0F80
        108F  2117
003869  1090  D380 145F              LNJ    $B5,<BLANKS              PRINT TWO BLANKS
003872                               FNERG  NULL
003873                               CALL   ZV$TH,ZV$THZ,SLDB+$AF-1,DCBS (REPORT DATA BUFFER LOCATION)
        1092  FBC0 0003
        1094  D380 0000      X
        1096  0F80
        1097  21B3
        1098  20F2
003874                               CALL   ZV$TH,ZV$THZ,STAT1        REPORT STATUS WORD 1
        1099  FBC0 0003
        109B  D380 0000      X
        109D  0F80
        109E  21B4
003875                               CALL   ZV$TH,ZV$THZ,STAT2        REPORT STATUS WORD 2
        109F  FBC0 0003
        10A1  D380 0000      X
        10A3  0F80
        10A4  21B5
003876                               *
003877  10A5  D380 100F              FNERC  LNJ    $B5,<SENSE           CHECK FOR "NEXT ?:"
003878  10A7  DC80 10AA              FNERD  LDB    $B5,<FNERB5
003879  10A9  8385                  JMP    $B5
003880                               *
003881  10AA  0000                  FNERB5 RESV   $AF,0                SAVED RETURN ADDRESS
003882                               ********************************************************************
003883                               * RESET RETRY AND DATA ERROR FLAGS
003884                               *
003885  10AB  8700 20F4              RSET   CL     <DEFL                CLEAR FLAGS
003886  10AD  8700 21A2                     CL     <RTFL
003887  10AF  1CF8                          LDV    $R1,=-8
003888  10B0  9F00 20F3                     STR    $R1,<DECN            COUNT UP TO 8 DATA ERRORS PER TRY
003889  10B2  D388 21A3                     LNJ    $B5,*<RTRN           DO ROUTINE PRIOR TO RETRY
003890  10B4  8388 21A4                     JMP    *<RTRY               GO TO RETRY LOCATION
003891                               ********************************************************************
003892                               * SET UP PRIOR TO READING OR DOING A TEST INVOLVING READING
003893                               *
003894  10B6  1CF8                  SET    LDV    $R1,=-8
003895  10B7  9F00 20F3                     STR    $R1,<DECN            ALLOW 8 DATA ERRORS
003896  10B9  1CFC                          LDV    $R1,=-4
003897  10BA  9F00 21A1                     STR    $R1,<RTCN            ALLOW 3 TRIES/RETRIES
003898  10BC  8700 20F4                     CL     <DEFL                DATA ERROR FLAG FOR FIRST ERROR
003899  10BE  8700 21A2                     CL     <RTFL                RETRY FLAAG
003900  10C0  8385                  JMP    $B5                  RETURN
003901                               ********************************************************************
003902                               *
003903                               * RETRY ROUTINES
003904                               *
003905                               *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
003906  10C1  8386                  RT2    JMP    $B6                  NOP ROUTINE FOR HANDLER
003907                               *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
003908  10C2  8385                  RT4    JMP    $B5                  DO NOTHING, JUST RETURN
003909                               *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
003910  10C3  DF80 10D0              RT5    STB    $B5,<RT5B5           (BSR/BSR/FSR/ERA)
003911  10C5  D380 142B                     LNJ    $B5,<BSRW
003912  10C7  D380 142B                     LNJ    $B5,<BSRW
003913  10C9  D380 13C1                     LNJ    $B5,<FSRW
003914  10CB  D380 134D                     LNJ    $B5,<ERAW
003915  10CD  DC80 10D0                     LDB    $B5,<RT5B5
003916  10CF  8385                  JMP    $B5
003917                               *
003918  10D0  0000                  RT5B5  RESV   $AF,0                RETURN ADDRESS
003919                               *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
003920  10D1  DF80 10DC              RT6    STB    $B5,<RT6B5           (BSR/BSR/FSR)
003921  10D3  D380 142B                     LNJ    $B5,<BSRW
003922  10D5  D380 142B                     LNJ    $B5,<BSRW
003923  10D7  D380 13C1                     LNJ    $B5,<FSRW
003924  10D9  DC80 10DC                     LDB    $B5,<RT6B5
003925  10DB  8385                  JMP    $B5
003926                               *
003927  10DC  0000                  RT6B5  RESV   $AF,0                RETURN ADDRESS
003928                               ********************************************************************
003929                               * ABORT ROUTINES
003930                               *
003931  10DD  8385                  ABR3   JMP    $B5                  (NOP)
003932                               ********************************************************************
003933                               * WRITE A RECORD
003934                               *
003935  10DE  9875                  WREC   LDR    $R1,+$B5             GET DATA
003936  10DF  9F00 2190                    STR    $R1,<PTRN            SAVE IT
003937  10E1  9875                         LDR    $R1,+$B5             GET RANGE
003938  10E2  9F00 219E                    STR    $R1,<RNG             SAVE IT
003939  10E4  9CF5                         LDB    $B1,+$B5             GET BUFFER ADDRESS
003940  10E5  9F80 21B3                    STB    $B1,<SLDB            START OF CURRENT DATA BUFFER
003941  10E7  9F80 12C2                    STB    $B1,<ISBSB           SB BUFFER PNTR FOR ISR CHECK
003942  10E9  9CF5                         LDB    $B1,+$B5             RETRY ROUTINE POINTER
003943  10EA  9F80 21A3                    STB    $B1,<RTRN
003944  10EC  9CF5                         LDB    $B1,+$B5             RETRY LOCATION POINTER
003945  10ED  9F80 21A4                    STB    $B1,<RTRY
003946  10EF  9CF5                         LDB    $B1,+$B5             ABORT ROUTINE POINTER
003947  10F0  9F80 20E6                    STB    $B1,<ABRN
003948  10F2  9CF5                         LDB    $B1,+$B5             ABORT LOCATION POINTER
003949  10F3  9F80 218B                    STB    $B1,<NTST
003950                               *
003951  10F5  DF80 114D                    STB    $B5,<WRECB5          RETURN ADDRESS
```

```
003952                                  *
003953                                  * PARAMETERS HAVE NOW BEEN PASSED ON. NEXT FILL BUFFER AND WRITE, ETC.
003954                                  *
003955  10F7  D380 0FEC                        LNJ     $B5,<INSTAT
003956  10F9  6280 21B4                        LB      <STAT1,=Z'8000'         RDY BIT
        10FB  8000
003957  10FC  0504                             BBT     >WRECA                  B IF READY
003958  10FD  D380 1030                        LNJ     $B5,<FNER               NOT RDY WHEN ABOUT TO WRITE
003959  10FF  3138                      FNER18 DC      '18'                    LABEL2
003960                                  *
003961  1100  D380 114E                 WRECA  LNJ     $B5,<FWB                FILL WRITE BUFFER
003962  1102  9080 2186                        LDH     $R1,<LABEL1             GET LEFT HALF OF LABEL1
003963  1104  1D41                             CMV     $R1,='A'                IS IT MODE "A"?
003964  1105  0980 1115                        BNE     <WRECB                  B IF NOT DATA TESTS
003965  1107  9800 2186                        LDR     $R1,<LABEL1
003966  1109  9970 4550                        CMR     $R1,='EP'               IS IT "END-PASS" ?
003967  110B  0900 1115                        BE      <WRECB                  B IF END-PASS, DON'T CHANGE DATA
003968                                  *
003969                                  * NOW STORE FILE AND RECORD NUMBERS IN FIRST FOUR BYTES
003970                                  *
003971  110D  9C80 21B3                        LDB     $B1,<SLDB               BUFFER ADDRESS
003972  110F  9830 20FE                        LDR     $R1,<DXFN,$R3           GET FILE NMBR
003973  1111  9F71                             STR     $R1,+$B1                FIRST TWO BYTES OF RECORD
003974  1112  9830 210E                        LDR     $R1,<DXRN,$R3           GET RECORD NMBR
003975  1114  9F71                             STR     $R1,+$B1                SECOND TWO BYTES OF RECORD
003976                                  *
003977  1115  D380 0FDB                 WRECB  LNJ     $B5,<RWAQ               MODE?
003978  1117  0F87                             B       >WRECC                  R, RETURN WITHOUT WRITING
003979  1118  0F88                             B       >WRECD                  W, WRITE
003980  1119  0F7F                             NOP     >$-1                    A
003981  111A  8280 20E7                        LB      <AUFL,=1                Q, CHECK IF AU TEST
        111C  0001
003982  111D  0583                             BBF     >WRECD                  B IF NOT AU, IE., WRITE
003983  111E  0F80 1148                 WRECC  B       <WRECI                  RETURN WITHOUT WRITING
003984                                  *
003985                                  * NOW WRITE THE DATA, WAIT FOR RDY AND CHECK STATUS
003986                                  *
003987  1120  8AB0 210E                 WRECD  INC     <DXRN,$R3               BUMP RECORD COUNTER
003988  1122  8188 21B3                 WRECE  IOLD    *<SLDB,<IOWRIT,<RNG
        1124  0000 22BC
        1126  0000 219E
003989  1128  07FA                             BIOF    >WRECE
003990  1129  8000 22A2                 WRECF  IO      <WRITE,<OTTAS<          WRITE
        112B  0000 22C2
003991  112D  07FC                             BIOF    >WRECF
003992                                  *
003993                                  * WRITE HAS BEGUN, WAIT FOR RDY, CHECK FOR STATUS ERRORS
003994                                  *
003995  112E  D380 177B                        LNJ     $B5,<WAIT               WAIT FOR RANGE RUNOUT
003996  1130  D380 0FEC                        LNJ     $B5,<INSTAT             GET STATUS
003997  1132  8800 21B4                        LBF     <STAT1,=Z'0100'         CHECK FOR EOT
        1134  0100
003998  1135  0584                             BBF     >WRECG                  B IF NOT EOT
003999  1136  8900 2112                        LBT     <EOTFLG,=1              SET EOT FLAG
        1138  0001
004000  1139  9800 21B4                 WRECG  LDR     $R1,<STAT1              GET FIRST STATUS WORD
004001  113B  9970 8000                        CMR     $R1,=Z'8000'            RDY
004002  113D  0988                             BNE     >WRECH                  B IF STATUS NG
004003  113E  9800 21B5                        LDR     $R1,<STAT2
004004  1140  9630 20FA                        XOR     $R1,<DXDENS,$R3         MASK WITH DENSITY BIT
004005  1142  9970 8000                        CMR     $R1,=Z'8000'            ON-LINE
004006  1144  0904                             BE      >WRECI                  B IF OK
004007  1145  D380 1030                 WRECH  LNJ     $B5,<FNER               STATUS NG AFTER A "WRITE" OPERATION
004008  1147  3135                      FNER15 DC      '15'                    LABEL2
004009                                  *
004010  1148  D380 100F                 WRECI  LNJ     $B5,<SENSE              NEXT?
004011  114A  DC80 114D                        LDB     $B5,<WRECB5
004012  114C  8385                             JMP     $B5
004013                                  *
004014  114D  0000                      WRECB5 RESV    $AF,0                   RETURN ADDRESS
004015                                  ********************************************************************
004016                                  * FILL WRITE BUFFER
004017                                  *
004018  114E  DF80 118C                 FWB    STB     $B5,<FWBB5
004019  1150  9C80 21B3                        LDB     $B1,<SLDB               BUFFER ADDRESS
004020  1152  9800 2190                        LDR     $R1,<PTRN               GET DATA PATTERN
004021  1154  9970 5244                        CMR     $R1,='RD'               RD = RANDOM DATA
004022  1156  0900 116F                        BE      <FWBB                   B IF RANDOM DATA REQUIRED
004023  1158  9970 00A9                        CMR     $R1,=X'A9'
004024  115A  0900 1167                        BE      <FWBF                   B IF CHECKER-BOARD PATTERN REQUIRED
004025  115C  9970 0056                        CMR     $R1,=X'56'
004026  115E  0900 1167                        BE      <FWBF                   B IF CHECKER-BOARD PATTERN REQUIRED
004027                                  *
004028                                  * FILL BUFFER WITH FIXED DATA
004029                                  *
004030  1160  8752                             CL      =$R2                    USE R2 AS INDEX THROUGH BUFFER
004031  1161  97ED                      FWBA   STH     $R1,$B1,+$R2            STORE THE DATA
004032  1162  A900 219E                        CMR     $R2,<RNG
004033  1164  027D                             BL      >FWBA                   KEEP FILLING
004034  1165  0F80 1183                        B       <FWBC                   DATA DONE, PAD REST OF BUFFER
004035                                  *
004036                                  * FILL WRITE BUFFER WITH CHECKER-BOARD PATTERN.
004037                                  *
004038  1167  8752                      FWBF   CL      =$R2                    CLEAR INDEX
004039  1168  97ED                      FWBG   STH     $R1,$B1,+$R2            STORE THE DATA
004040  1169  8651                             CPL     =$R1                    INVERT DATA TO FORM CHECKER-BOARD
004041  116A  A900 219E                        CMR     $R2,<RNG
004042  116C  027C                             BL      >FWBG                   B IF NOT FILLED YET
004043  116D  0F80 1183                        B       <FWBC                   DATA DONE, PAD REST OF BUFFER
004044                                  *
004045                                  * FILL WRITE BUFFER WITH RANDOM DATA
004046                                  *
004047  116F  9830 20FE                 FWBB   LDR     $R1,<DXFN,$R3           GET FILE NMBR
004048  1171  9A30 210E                        ADD     $R1,<DXRN,$R3           ADD REC NMBR
004049  1173  9F00 0000         X              STR     $R1,<ZV$FRB             SAVE AS RANDOM SEED
004050  1175  8752                             CL      =$R2                    CLEAR INDEX
004051                                  FWBE   CALL    ZV$FR,TEMPA,DC1         GET A RANDOM NMBR
        1176  FBC0 0003
        1178  D380 0000         X
        117A  0F80
        117B  21B7
        117C  20EC
004052  117D  9800 21B7                        LDR     $R1,<TEMPA
```

```
004053  117F  97ED                        STH    $R1,$B1,+$R2          STORE THE DATA
004054  1180  A900 219E                    CMR    $R2,<RNG
004055  1182  0274                        BL     >FWBE                 B IF NOT FILLED YET
004056                              *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
004057                              * NOW FILL REST OF BUFFER WITH ALL X'FF'
004058                              *
004059  1183  9870 00FF              FWBC  LDR    $R1,=X'FF'            GET DATA
004060  1185  97ED                  FWBD  STH    $R1,$B1,+$R2          STORE DATA
004061  1186  A970 0900                    CMR    $R2,=2*(1024+128)     BUFFER SIZE, BYTES
004062  1188  027D                        BL     >FWBD                 KEEP FILLING TILL DONE
004063                              *
004064  1189  DC80 118C                    LDB    $B5,<FWBB5            FORM RETURN ADDRESS
004065  118D  8385                        JMP    $B5                   RETURN
004066                              *
004067  118C  0000                  FWBB5 RESV   $AF,0                 RETURN ADDRESS
004068                              ********************************************************************
004069                              * READ A RECORD AND CHECK DATA
004070                              *
004071  118D  9875                  RREC  LDR    $R1,+$B5              GET BYTE RANGE
004072  118E  9F00 219E                    STR    $R1,<RNG              SAVE IT
004073  1190  9CF5                        LDB    $B1,+$B5              GET READ BUFFER ADR
004074  1191  9F80 21B3                    STB    $B1,<SLDB             SAVE IT
004075  1193  9F80 11AF                    STB    $B1,<RRECBB           ALSO FOR FILL ROUTINE
004076  1195  9CF5                        LDB    $B1,+$B5              POINTER TO...
004077  1196  9F80 21A3                    STB    $B1,<RTRN             RETRY ROUTINE
004078  1198  9CF5                        LDB    $B1,+$B5              POINTER TO...
004079  1199  9F80 21A4                    STB    $B1,<RTRY             RETRY LOCATION
004080  119B  9CF5                        LDB    $B1,+$B5              POINTER TO...
004081  119C  9F80 20E6                    STB    $B1,<ABRN             ABORT ROUTINE
004082  119E  9CF5                        LDB    $B1,+$B5              POINTER TO...
004083  119F  9F80 218B                    STB    $B1,<NTST             ABORT LOCATION (NEXT TEST)
004084                              *
004085  11A1  DF80 11CF                    STB    $B5,<RRECB5           SAVE RETURN ADDRESS
004086                              *
004087                              * PARAMETERS PASSED, NOW PRELOAD BUFFER, THEN READ AND CHECK.
004088                              *
004089  11A3  D380 0FEC                    LNJ    $B5,<INSTAT           GET STATUS
004090  11A5  8280 21B4                    LB     <STAT1,=Z'8000'
      11A7  8000
004091  11A8  0504                        BBT    >RRECB                B IF OK
004092  11A9  D380 1030                    LNJ    $B5,<FNER             SHOULD BE RDY PRIOR TO READ
004093  11AB  3136                  FNER16 DC     '16'                  LABEL2
004094                              *
004095  11AC  D380 0000         X   RRECB  LNJ    $B5,<ZV$F             FILL THE BUFFER
004096  11AE  0F84                        B      >RRECBA
004097  11AF  1C66                  RRECBB DC     <RBUF                 BUFFER
004098  11B0  21BF                        DC     <X3333                DATA
004099  11B1  2189                        DC     <LENGTH               RANGE
004100                              *
004101  11B2  8AB0 210E             RRECBA INC    <DXRN,$R3             INCREMENT RECORD COUNTER
004102  11B4  81C8 0FFE             RRECC  IOLD   *SLDB,<IOREAD,<RNG
      11B6  0000 22BB
      11B8  0000 219E
004103  11BA  07FA                        BIOF   >RRECC
004104  11BB  8000 229F             RRECD  IO     <READFD,<OUTASK       (START TO READ)
      11BD  0000 22C2
004105  11BF  07FC                        BIOF   >RRECD
004106                              *
004107                              * CHECK STATUS
004108                              *
004109  11C0  D380 177B                    LNJ    $B5,<WAIT             WAIT FOR READY
004110  11C2  D380 1480                    LNJ    $B5,<ERCK             CHECK STATUS ERRORS, ETC.
004111  11C4  D380 100F                    LNJ    $B5,<SENSE            NEXT?
004112                              *
004113                              * CHECK DATA
004114                              *
004115  11C6  9C80 12C2                    LDB    $B1,<ISBSB            POINTER TO SB (SAVED DURING WRITE)
004116  11C8  AC80 21B3                    LDB    $B2,<SLDB             POINTER TO "IS" BUFFER
004117  11CA  D380 123F                    LNJ    $B5,<ISB              CHECK THE DATA
004118                              *
004119  11CC  DC80 11CF                    LDB    $B5,<RRECB5           RESTORE RETURN ADDRESS
004120  11CE  8385                        JMP    $B5                   RETURN
004121                              *
004122  11CF  0000                  RRECB5 RESV   $AF,0                 RETURN ADDRESS
004123                              ********************************************************************
004124                              * WRITE A RECORD, BACKSPACE, READ IT AND CHECK DATA
004125                              *
004126  11D0  9875                  ONES  LDR    $R1,+$B5              GET DATA PATTERN
004127  11D1  9F00 11DC                    STR    $R1,<ONESB
004128  11D3  9875                        LDR    $R1,+$B5              GET BYTE RANGE
004129  11D4  9F00 11DD                    STR    $R1,<ONESC
004130  11D6  9F00 11F2                    STR    $R1,<ONESF
004131  11D8  DF80 1201                    STB    $B5,<ONESB5           SAVE RETURN ADDRESS
004132                              *
004133  11DA  D380 10DE             ONESA LNJ    $B5,<WREC             WRITE RECORD
004134  11DC  0000                  ONESB RESV   1,0                   DATA
004135  11DD  0000                  ONESC RESV   1,0                   BYTE RANGE
004136  11DE  17E6                        DC     <WBUF                 WRITE BUFFER
004137  11DF  10C3                        DC     <RT5                  RETRY ROUTINE (BSR/BSR/FSR/ERA)
004138  11E0  11DA                        DC     <ONESA                RETRY LOCATION
004139  11E1  10DD                        DC     <ABR3                 ABORT ROUTINE (NOP)
004140  11E2  11F8                        DC     <ONESG                ABORT LOCATION
004141                              *
004142                              * NOW CHECK MODE
004143                              *
004144  11E3  D380 0FDB                    LNJ    $B5,<RWAQ             MODE?
004145  11E5  0F89                        B      >ONESD                R, GO ON AND READ
004146  11E6  0F98                        B      >ONESH                W, RETURN, NO READ REQUIRED
004147  11E7  0F7F                        NOP    >$-1                  A, GO AHEAD WITH READ BUT FIRST...
004148  11E8  8280 20E7                    LB     <AUFL,=1              Q, SEE IF IN AU TEST
      11EA  0001
004149  11EB  0503                        BBT    >ONESD                B OVER BACKSPACE IF TEST AU
004150  11EC  D380 142B                    LNJ    $B5,<BSRW             BACKSPACE A RECORD, WAIT
004151                              *
004152                              * NOW READ AND CHECK DATA
004153                              *
004154  11EE  D380 10B6             ONESD LNJ    $B5,<SET              SET UP FOR READ
004155  11F0  D380 118D             ONESE LNJ    $B5,<RREC             READ A RECORD
004156  11F2  0000                  ONESF RESV   1,0                   BYTE RANGE
004157  11F3  1C66                        DC     <RBUF                 BUFFER ADDRESS
004158  11F4  10D1                        DC     <RT6                  RETRY ROUTINE (BSR/BSR/FSR)
004159  11F5  11F0                        DC     <ONESE                RETRY LOCATION
004160  11F6  10DD                        DC     <ABR3                 ABORT ROUTINE (NOP)
```

```
004161  11F7  11F8                    DC      <ONESG                  ABORT LOCATION
004162                          *
004163  11F8  8280 20E7         ONESG  LB      <AUFL,=1                CHECK IF TEST AU
        11FA  0001
004164  11FB  0583                    BBF     >ONESH                  B IF NOT AU TEST, RETURN
004165  11FC  D380 0B9B                LNJ     $B5,<AU6                RETURN TO TEST AU
004166                          *
004167  11FE  DC80 1201         ONESH  LDB     $B5,<ONESB5             SET UP FOR RETURN
004168  1200  8385                    JMP     $B5                     RETURN
004169                          *
004170  1201  0000              ONESB5 RESV    $AF,0                   RETURN ADDRESS
004171                          ********************************************************************
004172                          * WRITE, READ AND CHECK A RANDOM LENGTH RECORD OF RANDOM DATA
004173                          *
004174  1202  DF80 123E         RLD    STB     $B5,<RLDB5
004175  1204  9830 20FE         RLDA   LDR     $R1,<DXFN,$R3           FILE COUNT
004176  1206  9A30 210E                ADD     $R1,<DXRN,$R3           RECORD COUNT
004177  1208  9F00 0000       X        STR     $R1,<ZV$FRB             USE SUM AS BASE FOR RANDOM NMBR
004178                                 CALL    ZV$FR,WBUF,DC32         GET SOME RANDOM NUMBERS
        120A  FBC0 0003
        120C  D380 0000       X
        120E  0F80
        120F  17E6
        1210  20F0
004179  1211  9800 1805                LDR     $R1,<WBUF+31            GET AN ARBITRARY NUMBER
004180  1213  9570 07FF                AND     $R1,=2047               TRUNCATE TO 2047 BYTES
004181  1215  8AD1                    INC     =$R1                    FORM TRIAL RANGE, 1-2048 BYTES
004182  1216  1D12                    CMV     $R1,=18
004183  1217  0B62                    BAGE    >RLDB                   B IF 18 OR GREATER
004184  1218  1E12                    ADV     $R1,=18                 MAKE RANGE AT LEAST 18
004185  1219  9F00 1220       RLDB   STR     $R1,<RLDC               RANGE FOR WREC
004186  121B  9F00 1235                STR     $R1,<RLDF               RANGE FOR RREC
004187                          *
004188                          * RANDOM RANGE AND BASE JUST DONE, NOW WRITE RECORD
004189                          *
004190  121D  D380 10DE                LNJ     $B5,<WREC               WRITE A RECORD
004191  121F  5244                    DC      'RD'                    RANDOM DATA
004192  1220  0000              RLDC   RESV    1,0                     BYTE RANGE
004193  1221  17E6                    DC      <WBUF                   BUF ADR
004194  1222  10C3                    DC      <RTS                    RETRY ROUTINE (BSR/BSR/FSR/ERA)
004195  1223  1204                    DC      <RLDA                   RETRY LOCATION
004196  1224  10DD                    DC      <ABR3                   ABORT ROUTINE (NOP)
004197  1225  123B                    DC      <RLDG                   ABORT LOCATION (END OF TEST)
004198                          *
004199                          * CHECK FOR MODE
004200                          *
004201  1226  D380 0FDB                LNJ     $B5,<RWAQ               MODE?
004202  1228  0F89                    B       >RLDD                   R, CONTINUE WITH READ
004203  1229  0F92                    B       >RLDG                   W, RETURN, SKIP THE READ
004204  122A  0F7F                    NOP     >$-1                    A, SEE IF IN TEST AU
004205  122B  8280 20E7               LB      <AUFL,=1                Q, DITTO
        122D  0001
004206  122E  0503                    BBT     >RLDD                   B IF TEST AU, DON'T BACK-SPACE
004207  122F  D380 142B                LNJ     $B5,<BSRW               BACK A RECORD
004208                          *
004209                          * READ AND CHECK DATA
004210                          *
004211  1231  D380 10B6         RLDD   LNJ     $B5,<SET                SET UP FOR READ
004212  1233  D380 118D         RLDE   LNJ     $B5,<RREC               READ
004213  1235  0000              RLDF   RESV    1,0                     BYTE RANGE
004214  1236  1C66                    DC      <RBUF                   BUFFER
004215  1237  10D1                    DC      <RTS                    RETRY ROUTINE (BSR/BSR/FSR)
004216  1238  1233                    DC      <RLDE                   RETRY LOCATION
004217  1239  10DD                    DC      <ABR3                   ABORT ROUTINE (NOP)
004218  123A  123B                    DC      <RLDG                   ABORT LOCATION (END OF TEST)
004219                          *
004220  123B  DC80 123E         RLDG   LDB     $B5,<RLDB5              RESTORE RETURN ADDRESS
004221  123D  8385                    JMP     $B5                     RETURN
004222                          *
004223  123E  0000              RLDB5  RESV    $AF,0                   RETURN ADDRESS
004224                          ********************************************************************
004225                          * CHECK FOR END-OF-PASS AND CHECK DATA (B1->SB, B2->IS)
004226                          *
004227  123F  DF80 12C4         ISB    STB     $B5,<ISBB5              SAVE B5 FOR RETURN ADDRESS
004228  1241  9F80 12C2                STB     $B1,<ISBSB              POINTER TO "SHOULD-BE" DATA
004229  1243  AF80 12C3                STB     $B2,<ISBIS              POINTER TO "IS" DATA
004230  1245  9802                    LDR     $R1,$B2                 GET "IS" FIRST TWO BYTES
004231  1246  9942 0001                CMR     $R1,$B2.1               COMPARE TO NEXT TWO BYTES
004232  1248  0980 1261                BNE     <ISBB                   B IF NOT END-OF-PASS
004233  124A  9970 FFFF                CMR     $R1,=Z'FFFF'            COMPARE TO E-O-P INDICATOR
004234  124C  0980 1261                BNE     <ISBB                   B IF NOT E-O-P
004235  124E  9800 2186                LDR     $R1,<LABEL1
004236  1250  9970 4444                CMR     $R1,='DD'               CHECK IF DEBUG MODE, LINK "D"
004237  1252  0900 1261                BE      <ISBB                   B IF "DD" MODE
004238  1254  1048                    SOR     $R1,8
004239  1255  91F0 5420                CMH     $R1,='T'
004240  1257  0900 1261                BE      <ISBB                   B IF T1-T9
004241  1259  D380 0FDB                LNJ     $B5,<RWAQ               CHECK WHICH MODE
004242  125B  0F7F                    NOP     >$-1                    R
004243  125C  0F83                    B       >ISBA                   W
004244  125D  0F7F                    NOP     >$-1                    A
004245  125E  0F83                    B       >ISBB                   Q, CONTINUE WITH DATA CHECK
004246  125F  0F80 0BD4         ISBA   B       <NDPS                   END-OF-PASS DETECTED IN MODE R/W
004247                          *
004248                          * NOW CHECK DATA
004249                          *
004250  1261  8752              ISBB   CL      =$R2                    CLEAR INDEX THROUGH WBUF AND RBUF (ALSO RNG)
004251  1262  E2A8 12C2         ISBC   LLH     $R6,*<ISBSB.$R2         GET "SB" DATA
004252  1264  F2A8 12C3                LLH     $R7,*<ISBIS.$R2         GET "IS" DATA
004253  1266  E957                    CMR     $R6,=$R7
004254  1267  0900 126B                BE      <ISBD                   B IF COMPARISON OK
004255  1269  D380 1298         DATANG LNJ     $B5,<ISBG               DATA ERROR DETECTED, REPORT IT
004256                          *
004257  126B  8AD2              ISBD   INC     =$R2                    BUMP THE INDEX
004258  126C  A900 219E                CMR     $R2,<RNG
004259  126E  0200 1262                BL      <ISBC                   B IF RANGE NOT YET DONE
004260                          *
004261                          * RANGE DONE, CHECK FOR DMA OVERFLOW. IF NO ERRORS, RETURN
004262                          *
004263  1270  92A2                    LLH     $R1,$B2.$R2             GET "IS" DATA FOR NEXT LOC AFTER BUF
004264  1271  1D33                    CMV     $R1,=X'33'              CHECK IF SAME AS BEFORE
004265  1272  0900 1277                BE      <ISBE                   B IF NO DMA OVERFLOW
004266  1274  D380 1030                LNJ     $B5,<FNER               DMA OVERFLOW
```

```
004267  1276  444F                FNERDO DC     'DO'                LABEL2
004268                             *
004269  1277  8280 20F4            ISBE   LB     <DEFL,=1
        1279  0001
004270  127A  0500 127F                   BBT    <ISBF               B IF DATA ERR FLAG SET
004271  127C  DC80 12C4                   LDB    $B5,<ISBB5          NO ERROR FOUND...
004272  127E  8385                       JMP    $B5                 RETURN
004273                             *
004274                             * IF ERRORS NOT SUPPRESSED AND NOT IN DEBUG MODE AND THE RETRY
004275                             * ROUTINE IS NOT "NOP", THEN PRINT MESSAGE "RETRY".
004276                             *
004277  127F  9800 2186            ISBF   LDR    $R1,<LABEL1         CHECK IF DEBUG MODE
004278  1281  9970 4444                   CMR    $R1,='DD'
004279  1283  0900 10AB                   BE     <RSET               B IF IN DEBUG COMPARE LINK
004280  1285  8280 21B6                   LB     <SUPRES,=1
        1287  0001
004281  1288  0500 10AB                   BBT    <RSET               B IF ERRORS SUPPRESSED
004282  128A  9B80 10C2                   LAB    $B1,<RT4            ADDRESS OF "NOP" ROUTINE
004283  128C  9D80 21A3                   CMB    $B1,<RTRN           COMPARE TO RETRY ROUTINE ADDRESS
004284  128E  0900 10AB                   BE     <RSET               B IF RETRY IS A "NOP"
004285                                     CALL   ZV$T,MSRTRY         PRINT "RETRY"
        1290  FBC0 0003
        1292  D380 0000      X
        1294  0F80
        1295  228C
004286  1296  0F80 10AB                   B      <RSET
004287                             *
004288                             * REPORT DATA ERRORS
004289                             *
004290  1298  0F80 12C5            ISBG   STB    $B5,<ISBGB5         SAVE B5 FOR RETURN
004291  129A  FB80 1269                   LAB    $B7,<DATANG         LOCATION WHERE ERROR DETECTED
004292  129C  FF80 107C                   STB    $B7,<FNERF
004293  129E  9870 4441            FNERDA LDR    $R1,='DA'           LABEL2 FOR DATA ERROR
004294  12A0  9F00 2187                   STR    $R1,<LABEL2
004295  12A2  8900 21A2                   LBT    <RTFL,=1            SET RETRY FLAG
        12A4  0001
004296  12A5  0500 12C6                   BBT    <DERA               B IF RTFL WAS SET
004297  12A7  8A80 21A1                   INC    <RTCN               RETRY COUNTER
004298  12A9  8980 21A1                   CMZ    <RTCN
004299  12AB  0800 12C6                   BAL    <DERA               B IF 3 TRIES NOT DONE YET
004300  12AD  8280 21B6                   LB     <SUPRES,=1
        12AF  0001
004301  12B0  0500 12B8                   BBT    <ISBH               B IF REPORTS SUPPRESSED
004302                                     CALL   ZV$T,MSABRT         PRINT MESSAGE "ABORT"
        12B2  FBC0 0003
        12B4  D380 0000      X
        12B6  0F80
        12B7  2218
004303  12B8  9B80 10C2            ISBH   LAB    $B1,<RT4
004304  12BA  9F80 21A3                   STB    $B1,<RTRN           SET RETRY ROUTINE TO "NOP"
004305  12BC  D380 10B6                   LNJ    $B5,<SET            SET UP FOR 8 DATA ERRS, 3 TRIES
004306  12BE  D388 20E6                   LNJ    $B5,*<ABRN          DO THE ABORT ROUTINE
004307  12C0  8388 218B                   JMP    *<NIST              GO ON TO NEXT TEST
004308                             *
004309                             *
004310  12C2  0000                 ISBSB  RESV   $AF,0               POINTER TO "SHOULD-BE" DATA
004311  12C3  0000                 ISBIS  RESV   $AF,0               POINTER TO "IS" DATA
004312  12C4  0000                 ISBB5  RESV   $AF,0               HOLD B5 FOR RETURN FROM "ISB"
004313  12C5  0000                 ISBGB5 RESV   $AF,0               HOLD B5 FOR RETURN FROM "ISBG"
004314                             ********************************************************************
004315                             * DIDN'T ABORT. REPORT THE ERROR AND DO RETRY IF NECESSARY.
004316                             *
004317  12C6  8900 20F4            DERA   LBT    <DEFL,=1            SET DATA ERROR FLAG
        12C8  0001
004318  12C9  0500 12CF                   BBT    <DERB               B IF FLAG SET
004319  12CB  D380 1039                   LNJ    $B5,<RPRT           PRINT ERR REPORT IF FIRST DATA ERR ONLY
004320  12CD  0F80 12D5                   B      <DERC
004321                             *
004322  12CF  8A80 20F3            DERB   INC    <DECN               COUNT DATA ERRORS (FROM -8)
004323  12D1  8980 20F3                   CMZ    <DECN
004324  12D3  0880 127F                   BAGE   <ISBF               B IF TOO MANY DATA ERRORS
004325                             *
004326                             * REPORT IS-SB, CONTINUE DATA CHECK
004327                             *
004328  12D5  D380 12DA            DERC   LNJ    $B5,<PISB           PRINT "IS-SB" INFO
004329  12D7  DC80 12C5                   LDB    $B5,<ISBGB5
004330  12D9  8385                       JMP    $B5                 RETURN
004331                             ********************************************************************
004332                             * PRINT "IS" AND "SHOULD-BE" DATA
004333                             *
004334  12DA  0F80 1319            PISB   STB    $B5,<PISBB5         SAVE B5 FOR RETURN
004335  12DC  8280 21B6                   LB     <SUPRES,=1
        12DE  0001
004336  12DF  0580 12E2                   BBF    <PISBA              B IF NOT SUPPRESSED
004337  12E1  8385                       JMP    $B5                 SUPPRESSED, RETURN
004338                             *
004339                             PISBA  CALL   ZV$T,ZV$TC,MSBYTE    PRINT "BYTE"
        12E2  FBC0 0003
        12E4  D380 0000      X
        12E6  0F80
        12E7  221C
004340  12E8  AF00 21B7                   STR    $R2,<TEMPA
004341                                     CALL   ZV$TH,TEMPA         PRINT HEX BYTE NUMBER
        12EA  FBC0 0003
        12EC  D380 0000      X
        12EE  0F80
        12EF  21B7
004342  12F0  8D00 21B7                   SDI    <TEMPA
004343                                     CALL   ZV$T,MSSB           PRINT "SB"
        12F2  FBC0 0003
        12F4  D380 0000      X
        12F6  0F80
        12F7  2290
004344                                     CALL   ZV$HA,ZV$HZ,TEMPA,VRBL CONVERT "SB" DATA ASCII
        12F8  FBC0 0003
        12FA  D380 0000      X
        12FC  0F80
        12FD  21B7
        12FE  21BB
004345  12FF  9800 21BD                   LDR    $R1,<VRBL+2
004346  1301  D380 0FCB                   LNJ    $B5,<VDTR           PRINT HEX "SB" DATA
004347                             *
004348                                     CALL   ZV$HA,ZV$T,MSIS     PRINT "IS"
```

```
         1303  FBC0 0003
         1305  D380 0000      A
         1307  0F80
         1308  2257
004349                        CALL   ZV$MA,ZV$MZ,TEMPB,VRBL CONVERT IS DATA TO ASCII
         1309  FBC0 0003
         130B  D380 0000      A
         130D  0F80
         130E  21BB
         130F  21BB
004350   1310  9800 21BD            LDR    $R1,<VRBL+2
004351   1312  D380 0FCB            LNJ    $B5,<VDTR              PRINT HEX "IS" DATA
004352                        *
004353   1314  D380 100F            LNJ    $B5,<SENSE             "NEXT ?"
004354   1316  DC80 1319            LDB    $B5,<PISBB5
004355   1318  8385                JMP    $B5                    RETURN
004356                        *
004357   1319  0000           PISBB5 RESV   $AF,0
004358                        ********************************************************************
004359                        * WRITE A FILE MARK AND WAIT TILL DONE
004360                        *
004361   131A  0F80 1340      WFM    STB    $B5,<WFMB5
004362   131C  9600 2166            LDR    $R1,<LABEL1
004363   131E  9970 4154            CMR    $R1,='AT'              ARE WE IN TEST "AT" ?
004364   1320  098A                BNE    >WFME                  B IF NOT "AT", LEAVE LVL ALONE
004365   1321  9630 20FE            LDR    $R1,<DXFN,$R3          GET FILE COUNT
004366   1323  1D08                CMV    $R1,=8
004367   1324  0986                BNE    >WFME                  B IF NOT GOING FROM 8 TO 9
004368   1325  8070 000A      WFMD  IO     =10,<OTRUPT            SET DEVICE LVL TO RUPT
         1327  0000 22BF
004369   1329  07FC                BIOF   >WFMD
004370   132A  D380 0FDB      WFME   LNJ    $B5,<RWAQ
004371   132C  0F8F                B      >WFMB                  R, DON'T WRITE ANYTHING
004372   132D  0F7F                NOP    >$-1                   W
004373   132E  0F7F                NOP    >$-1                   A
004374   132F  8AB0 20FE            INC    <DXFN,$R3              Q OR D, ADVANCE FILE COUNTER
004375   1331  8730 210E            CL     <DXRN,$R3              CLEAR RECORD COUNTER
004376   1333  8000 22A1            IO     <WRITFM,<OTTASK        WRITE FILE MARK
         1335  0000 22C2
004377   1337  D380 177B            LNJ    $B5,<WAIT              WAIT TILL DONE
004378   1339  0F80 133D            B      <WFMC                  RETURN
004379   133B  D380 1375      WFMB   LNJ    $B5,<FSFW              FORWARD SPACE A FILE
004380   133D  DC80 1340      WFMC   LDB    $B5,<WFMB5
004381   133F  8385                JMP    $B5                    RETURN
004382                        *
004383   1340  0000           WFMB5  RESV   $AF,0
004384                        ********************************************************************
004385                        * SET ALL CHANNEL CONSTANTS TO CURRENT DRIVE ADDRESS (PORT NUMBER)
004386                        *
004387   1341  9B80 22BA      SETCHN LAB    $B1,<CHAN              START OF CHAN ADDRESSES
004388   1343  AB80 22D0            LAB    $B2,<CHANZ             END OF CHAN ADDRESSES
004389   1345  9800 20F5            LDR    $R1,<DRIVE             DRIVE ADDRESS
004390   1347  9AF1           SETCHA SKM    $R1,+$B1,=Z'0180'      ADD DRIVE ADDRESS TO CHANNEL
         1348  0180
004391   1349  9DD2                CMB    $B1,=$B2
004392   134A  0200 1347            BL     <SETCHA                B IF TABLE NOT DONE YET
004393   134C  8385                JMP    $B5                    RETURN
004394                        ********************************************************************
004395                        * ERASE A BLOCK AND WAIT TILL DONE. CHECK STATUS.
004396                        *
004397   134D  0F80 136D      ERAW   STB    $B5,<ERAWB5
004398   134F  8000 22A0            IO     <ERASE,<OTTASK         ERASE TAPE
         1351  0000 22C2
004399   1353  D380 177B            LNJ    $B5,<WAIT
004400   1355  D380 0FEC            LNJ    $B5,<INSTAT            GET STATUS
004401   1357  9800 21B4            LDR    $R1,<STAT1             GET STATUS
004402   1359  8851                LBF    =$R1,=Z'0100'          DON'T CARE - EOT
         135A  0100
004403   135B  9970 8000            CMR    $R1,=Z'8000'           RDY,
004404   135D  0980 1367            BNE    <ERAWB                 B IF NG
004405   135F  9800 21B5            LDR    $R1,<STAT2             GET STATUS
004406   1361  9630 20FA            XOR    $R1,<DXDENS,$R3        MASK WITH DENSITY BIT
004407   1363  9970 8000            CMR    $R1,=Z'8000'
004408   1365  0900 136A            BE     <ERAWC                 B IF OK
004409   1367  D380 1030      ERAWB  LNJ    $B5,<FNER              STATUS NG AFTER ERASE
004410   1369  4531           FNERE1 DC     'E1'                   LABEL2
004411   136A  DC80 136D      ERAWC  LDB    $B5,<ERAWB5
004412   136C  8385                JMP    $B5
004413                        *
004414   136D  0000           ERAWB5 RESV   $AF,0
004415                        ********************************************************************
```

```
004416                          * WAIT FOR N/BSY, GET RANGE. HOWEVER, DON'T ALLOW FOR INTERRUPTS AS IN
004417                          * SUBROUTINE "WAIT".
004418                          *
004419  136E  8000 219D        GETRNG IO       <RANGE,<INRANG          GET RANGE
        1370  0000 22CA
004420  1372  0780 136E                BIOF    <GETRNG                 WAIT FOR N/BSY, GET RANGE REMAINING
004421  1374  8385                    JMP     $B5                     RETURN
004422                          ********************************************************************
004423                          * FORWARD SPACE A FILE AND WAIT TILL DONE
004424                          *
004425  1375  DF80 139D        FSFW    STB     $B5,<FSFWB5
004426  1377  D380 13B3                LNJ     $B5,<FSF                START TO SPACE
004427  1379  D380 177B                LNJ     $B5,<WAIT
004428  137B  D360 0FEC                LNJ     $B5,<INSTAT             GET STATUS
004429  137D  9800 21B4                LDR     $R1,<STAT1              GET STAT
004430  137F  8851                    LBF     =$R1,=Z'0100'           DON'T CARE - EOT
        1380  0100
004431  1381  9970 8400                CMR     $R1,=Z'8400'
004432  1383  0980 1397                BNE     <FSFWA                  B IF STAT1 NG
004433  1385  9800 21B5                LDR     $R1,<STAT2              GET STAT
004434  1387  9630 20FA                XOR     $R1,<DXDENS,$R3         MASK WITH DENSITY BIT
004435  1389  A0B0 2102                LDH     $R2,<DXMD,$R3           GET MODE
004436  138B  A1F0 5220                CMH     $R2,='R '
004437  138D  0904                    BE      >FSFWD                  B IF MODE "R"
004438  138E  A1F0 4420                CMH     $R2,='D '
004439  1390  0983                    BNE     >FSFWC                  B IF NOT MODE "D"
004440  1391  8851            FSFWD   LBF     =$R1,=Z'2000'           IGNORE "PROTECT" STATUS BIT
        1392  2000
004441  1393  9970 8000        FSFWC   CMR     $R1,=Z'8000'
004442  1395  0900 139A                BE      <FSFWB                  B IF STAT2 OK
004443  1397  D380 1030        FSFWA   LNJ     $B5,<FNER               STATUS NG AFTER FSFW
004444  1399  4631            FNERF1  DC      'F1'                    LABEL2
004445  139A  DC80 139D        FSFWB   LDB     $B5,<FSFWB5
004446  139C  8385                    JMP     $B5                     RETURN
004447                          *
004448  139D  0000            FSFWB5  RESV    $AF,0
004449                          ********************************************************************
004450                          * FORWARD SPACE A FILE, PAST E-O-T IF NECESSARY
004451                          *
004452  139E  DF80 13B2        FSFEOT  STB     $B5,<FSFEB5
004453  13A0  D380 13C1        FSFEA   LNJ     $B5,<FSRW
004454  13A2  8280 21B4                LB      <STAT1,=Z'0400'
        13A4  0400
004455  13A5  05FB                    BBF     >FSFEA                  B IF NOT YET FM DET
004456  13A6  9830 20FE                LDR     $R1,<DXFN,$R3           GET FILE COUNT
004457  13A8  1D08                    CMV     $R1,=8
004458  13A9  0A86                    BALE    >FSFEC                  B IF IN FIRST 8 FILES
004459  13AA  8070 000A        FSFED   IO      =10,<OTRUPT             SET UP FOR RUPTS
        13AC  0000 22BF
004460  13AE  07FC                    BIOF    >FSFED
004461  13AF  DC80 13B2        FSFEC   LDB     $B5,<FSFEB5
004462  13B1  8385                    JMP     $B5                     RETURN
004463                          *
004464  13B2  0000            FSFEB5  RESV    $AF,0
004465                          ********************************************************************
004466                          * START TO FORWARD SPACE A FILE
004467                          *
004468  13B3  DF80 13C0        FSF     STB     $B5,<FSFB5
004469  13B5  8AB0 20FE                INC     <DXFN,$R3               INCREMENT FILE COUNT
004470  13B7  8730 210E                CL      <DXRN,$R3               CLR RECORD COUNT
004471  13B9  8000 229D                IO      <FWDFIL,<OTTASK
        13BB  0000 22C2
004472  13BD  DC80 13C0                LDB     $B5,<FSFB5
004473  13BF  8385                    JMP     $B5                     RETURN
004474                          *
004475  13C0  0000            FSFB5   RESV    $AF,0
004476                          ********************************************************************
004477                          * FORWARD SPACE A RECORD AND WAIT TILL DONE
004478                          *
004479  13C1  DF80 13E8        FSRW    STB     $B5,<FSRWB5
004480  13C3  D380 13E9                LNJ     $B5,<FSR                START TO SPACE
004481  13C5  D380 177B                LNJ     $B5,<WAIT
004482  13C7  D360 0FEC                LNJ     $B5,<INSTAT             GET STAT
004483  13C9  9800 21B4                LDR     $R1,<STAT1              GET STAT
004484  13CB  8851                    LBF     =$R1,=Z'0500'           DON'T CARE - EOT, FM DET
        13CC  0500
004485  13CD  9970 8000                CMR     $R1,=Z'8000'
004486  13CF  0980 13D9                BNE     <FSRWA                  B IF STAT1 NG
004487  13D1  9800 21B5                LDR     $R1,<STAT2              GET STAT
004488  13D3  9630 20FA                XOR     $R1,<DXDENS,$R3         MASK WITH DENSITY BIT
004489  13D5  9970 8000                CMR     $R1,=Z'8000'            CHECK STAT2
004490  13D7  0900 13DC                BE      <FSRWB                  B IF STAT2 OK
004491  13D9  D380 1030        FSRWA   LNJ     $B5,<FNER               STAT NG AFTER FSRW
004492  13DB  5231            FNERR1  DC      'R1'                    LABEL2
004493  13DC  8280 21B4        FSRWB   LB      <STAT1,=Z'0400'         CHECK FOR FILE MARK
        13DE  0400
004494  13DF  05B0 13E5                BBF     <FSRWC                  B IF NO FILE MARK
004495  13E1  8AB0 20FE                INC     <DXFN,$R3               INCREMENT FILE COUNTER
004496  13E3  8730 210E                CL      <DXRN,$R3               CLR RECORD COUNTER
004497  13E5  DC80 13E8        FSRWC   LDB     $B5,<FSRWB5
004498  13E7  8385                    JMP     $B5                     RETURN
004499                          *
004500  13E8  0000            FSRWB5  RESV    $AF,0
004501                          ********************************************************************
004502                          * START TO FORWARD SPACE A RECORD
004503                          *
004504  13E9  DF80 13F4        FSR     STB     $B5,<FSRB5
004505  13EB  8AB0 210E                INC     <DXRN,$R3               BUMP RECORD COUNTER
004506  13ED  8000 229B                IO      <FWDREC,<OTTASK
        13EF  0000 22C2
004507  13F1  DC80 13F4                LDB     $B5,<FSRB5
004508  13F3  8385                    JMP     $B5                     RETURN
004509                          *
004510  13F4  0000            FSRB5   RESV    $AF,0
004511                          ********************************************************************
004512                          * BACKSPACE A FILE AND WAIT TILL DONE
004513                          *
004514  13F5  DF80 1413        BSFW    STB     $B5,<BSFWB5
004515  13F7  D380 1414                LNJ     $B5,<BSF                START TO SPACE
004516  13F9  D380 177B                LNJ     $B5,<WAIT
004517  13FB  D380 0FEC                LNJ     $B5,<INSTAT             GET STATUS
004518  13FD  9800 21B4                LDR     $R1,<STAT1              GET STAT
004519  13FF  8851                    LBF     =$R1,=Z'0300'           DON'T CARE - EOT, BOT
```

```
                1400  0300
004520  1401  9970  8400                CMR   $R1,=Z'8400'
004521  1403  0980  140D                BNE   <BSFWA                  B IF STAT1 NG
004522  1405  9800  21B5                LDR   $R1,<STAT2              GET STAT
004523  1407  9630  20FA                XOR   $R1,<DXDENS.$R3         MASK WITH DENSITY BIT
004524  1409  9970  8000                CMR   $R1,=Z'8000'
004525  140B  0900  1410                BE    <BSFWC                  B IF STAT2 OK
004526  140D  D380  1030    BSFWA       LNJ   $B5,<FNER               STATUS NG AFTER BSFW
004527  140F  4634          FNERF4      DC    'F4'                    LABEL2
004528  1410  DC80  1413    BSFWC       LDB   $B5,<BSFWB5
004529  1412  8385                      JMP   $B5                     RETURN
004530                      *
004531  1413  0000          BSFWB5      RESV  $AF,0
004532                      ********************************************************************
004533                      * START TO BACKSPACE A FILE
004534                      *
004535  1414  DF80  142A    BSF         STB   $B5,<BSFB5
004536  1416  8BB0  20FE                DEC   <DXFN.$R3               DEC FILE COUNTER
004537  1418  8730  210E                CL    <DXRN.$R3               CLR RECORD COUNTER
004538  141A  9830  20FE                LDR   $R1,<DXFN.$R3           GET FILE COUNT
004539  141C  1D08                      CMV   $R1,=8
004540  141D  0986                      BNE   >BSFB                   B IF NOT ARRIVING AT FILE 8
004541  141E  8070  0000    BSFA        IO    =0,<OTRUPT              NOW AT FILE 8, INHIBIT RUPTS
                1420  0000  22BF
004542  1422  07FC                      BIOF  >BSFA
004543  1423  8000  229E    BSFB        IO    <REVFIL,<OTTASK         START TO BACK SPACE
                1425  0000  22C2
004544  1427  DC80  142A                LDB   $B5,<BSFB5
004545  1429  8385                      JMP   $B5                     RETURN
004546                      *
004547  142A  0000          BSFB5       RESV  $AF,0
004548                      ********************************************************************
004549                      * BACKSPACE A RECORD AND WAIT TILL DONE
004550                      *
004551  142B  DF80  1452    BSRW        STB   $B5,<BSRWB5
004552  142D  D380  1453                LNJ   $B5,<BSR                START TO SPACE
004553  142F  D380  177B                LNJ   $B5,<WAIT
004554  1431  D380  0FEC                LNJ   $B5,<INSTAT             GET STAT
004555  1433  9800  21B4                LDR   $R1,<STAT1              GET STAT
004556  1435  8851                      LBF   =$R1,=Z'0700'           DON'T CARE - FM, BOT, EOT
                1436  0700
004557  1437  9970  8000                CMR   $R1,=Z'8000'
004558  1439  0980  1443                BNE   <BSRWA                  B IF STAT1 NG
004559  143B  9800  21B5                LDR   $R1,<STAT2              GET STAT
004560  143D  9630  20FA                XOR   $R1,<DXDENS.$R3         MASK WITH DENSITY BIT
004561  143F  9970  8000                CMR   $R1,=Z'8000'
004562  1441  0900  1446                BE    <BSRWC                  B IF STAT2 OK
004563  1443  D380  1030    BSRWA       LNJ   $B5,<FNER               STATUS NG AFTER BSRW
004564  1445  5234          FNERR4      DC    'R4'                    LABEL2
004565  1446  8280  21B4    BSRWC       LB    <STAT1,=Z'0400'         CHECK FOR FILE MARK
                1448  0400
004566  1449  0580  144F                BBF   <BSRWE                  B IF NO FILE MARK
004567  144B  8BB0  20FE                DEC   <DXFN.$R3               DECREMENT FILE CNTR IF FILE DETECTED
004568  144D  8730  210E    BSRWD       CL    <DXRN.$R3               CLR RECORD COUNTER
004569  144F  DC80  1452    BSRWE       LDB   $B5,<BSRWB5
004570  1451  8385                      JMP   $B5
004571                      *
004572  1452  0000          BSRWB5      RESV  $AF,0
004573                      ********************************************************************
004574                      * START TO BACKSPACE A RECORD
004575                      *
004576  1453  DF80  145E    BSR         STB   $B5,<BSRB5
004577  1455  8BB0  210E                DEC   <DXRN.$R3               DECREMENT RECORD COUNTER
004578  1457  8000  229C                IO    <REVREC,<OTTASK         START TO BACKSPACE
                1459  0000  22C2
004579  145B  DC80  145E                LDB   $B5,<BSRB5
004580  145D  8385                      JMP   $B5
004581                      *
004582  145E  0000          BSRB5       RESV  $AF,0
004583                      ********************************************************************
004584                      * PRINT TWO (2) BLANKS
004585                      *
004586  145F  DF80  1468    BLANKS      STB   $B5,<BLANB5
004587  1461  9870  2020                LDR   $R1,='  '
004588  1463  D380  0FCB                LNJ   $B5,<VDTR               PRINT THE BLANKS
004589  1465  DC80  1468                LDB   $B5,<BLANB5
004590  1467  8385                      JMP   $B5
004591                      *
004592  1468  0000          BLANB5      RESV  $AF,0
004593                      ********************************************************************
004594                      * CHECK THAT CURRENT UNIT IS ACTIVE AND NOT BUSY
004595                      *
004596  1469  DFC0  0015    ANB         STB   $B5,ANBB5
004597  146B  8000  21B4                IO    <STAT1,<INSTW1          TRY TO GET STATUS
                146D  0000  22CC
004598  146F  0700  1474                BIOT  <ANBA                   B IF N/BSY
004599  1471  D380  1030                LNJ   $B5,<FNER               SHOULDN'T BE BSY
004600  1473  3031          FNER01      DC    '01'                    LABEL2
004601  1474  8280  21B4    ANBA        LB    <STAT1,=Z'8000'
                1476  8000
004602  1477  0500  147C                BBT   <ANBB                   B IF RDY
004603  1479  D380  1030                LNJ   $B5,<FNER               SHOULD HAVE BEEN RDY
004604  147B  3032          FNER02      DC    '02'                    LABEL2
004605  147C  DC80  147F    ANBB        LDB   $B5,<ANBB5
004606  147E  8385                      JMP   $B5
004607                      *
004608  147F  0000          ANBB5       RESV  $AF,0
004609                      ********************************************************************
004610                      * CHECK FOR ERRORS
004611                      *
004612  1480  DF80  14E2    ERCK        STB   $B5,<ERCKB5
004613  1482  9800  2186                LDR   $R1,<LABEL1             GET LABEL
004614  1484  9970  4452                CMR   $R1,='DR'               "READ" MODE
004615  1486  0988                      BNE   >ERCKA                  B IF NOT READ MODE
004616  1487  8980  219F                CMZ   <RNGFLG
004617  1489  0905                      BE    >ERCKA                  SUPPRESS "LENGTH CHECKS"
004618  148A  8280  21B4                LB    <STAT1,=Z'3A7F'         IGNORE LENGTH CHECK
                148C  3A7F
004619  148D  0F84                      B     >ERCKB
004620  148E  8280  21B4    ERCKA       LB    <STAT1,=Z'3AFF'
                1490  3AFF
004621  1491  0500  14BC    ERCKB       BBT   <ERCKC                  B IF STAT1 NG
004622  1493  9800  2186                LDR   $R1,<LABEL1             GET LABEL
```

```
004623  1495  9970 444F                    CMR   $R1,='DO'
004624  1497  0900 14B3                    BE    <ERCKG                  B IF PROTECT STATUS IS OK
004625  1499  9970 4452                    CMR   $R1,='DR'
004626  149B  0900 14B3                    BE    <ERCKG
004627  149D  9970 4454                    CMR   $R1,='DT'
004628  149F  0900 14B3                    BE    <ERCKG
004629  14A1  9970 4455                    CMR   $R1,='DU'
004630  14A3  0900 14B3                    BE    <ERCKG
004631  14A5  9970 4456                    CMR   $R1,='DV'
004632  14A7  0900 14B3                    BE    <ERCKG
004633  14A9  9970 4459                    CMR   $R1,='DY'
004634  14AB  0900 14B3                    BE    <ERCKG
004635  14AD  90B0 2102                    LDH   $R1,<DXMD,$R3           GET CURRENT MODE
004636  14AF  91F0 5220                    CMH   $R1,='R '               MODE "R"
004637  14B1  0980 14B7                    BNE   <ERCKH                  B IF PROTECT STATUS IS NG
004638  14B3  8280 21B5           ERCKG    LB    <STAT2,=Z'4FFF'         IGNORE "PROTECT STATUS"
        14B5  4FFF
004639  14B6  0F84                        B     >ERCKI
004640  14B7  8280 21B5           ERCKH    LB    <STAT2,=Z'6FFF'         IGNORE ON-LINE AND HI-DENSITY
        14B9  6FFF
004641  14BA  0580 14BF           ERCKI    BBF   <ERCKD                  B IF STAT2 OK
004642  14BC  D380 1030           ERCKC    LNJ   $B5,<FNER               STATUS 1 AND/OR 2 NG
004643  14BE  3037                FNER07   DC    '07'                    LABEL2
004644                                     *
004645  14BF  8800 21A2           ERCKD    LBF   <RTFL,=1
        14C1  0001
004646  14C2  0500 14C7                    BBT   <ERCKE                  B IF RTFL SET, DO RETRY ROUTINE
004647  14C4  DC80 14E2                    LDB   $B5,<ERCKB5             RESTORE RETURN ADDRESS
004648  14C6  8385                        JMP   $B5                     RETURN FROM ERR CHECK, NO RETRY
004649                                     *
004650                                     * A RETRY IS IN ORDER. FIRST DO RETRY ROUTINE, THEN GO TO RETRY LOCATION.
004651                                     *
004652  14C7  D388 21A3           ERCKE    LNJ   $B5,*<RTRN              DO RETRY ROUTINE TO SET UP FOR RETRY
004653  14C9  9080 2186                    LDH   $R1,<LABEL1             CHECK IF MODE D (LABEL1 = "DX")
004654  14CB  91F0 4420                    CMH   $R1,='D '
004655  14CD  0900 14E0                    BE    <ERCKF                  B IF MODE D, DON'T PRINT RETRY
004656  14CF  8280 21B6                    LB    <SUPRES,=1
        14D1  0001
004657  14D2  0500 14E0                    BBT   <ERCKF                  IF SUPPRESSED, DON'T PRINT "RETRY"
004658  14D4  9B80 10C2                    LAB   $B1,<RT4                A "NOP" ROUTINE
004659  14D6  9D80 21A3                    CMB   $B1,<RTRN               RETRY ROUTINE
004660  14D8  0900 14E0                    BE    <ERCKF                  B IF RTRN IS "NOP", DON'T PRINT "RETRY"
004661                                     CALL  ZV$T,MSRTRY             PRINT "RETRY"
        14DA  FBC0 0003
        14DC  D380 0000       X
        14DE  0F80
        14DF  226C
004662  14E0  8388 21A4           ERCKF    JMP   *<RTRY                  GO TO RETRY LOCATION AND CONTINUE
004663                                     *
004664  14E2  0000                ERCKB5   RESV  $AF,0
004665                                     ********************************************************************************
004666                                     * CONSTANTS AND SAVE AREAS FOR TEST RP.
004667                                     *
004668  14E3  0080                LVQC     DC    Z'0080'
004669  14E4  0000                DEVSEM   RESV  1,0
004670  14E5  0000                CPCHAN   RESV  1,0
004671  14E6  0000                DLVL     RESV  1,0                     DEVICE LEVEL
004672  14E7  0000                CLVL     RESV  1,0                     *********
004673  14E8  0000                PLVL     RESV  1,0                     PROGRAM LEVEL
004674  14E9        003F      X   H00BF    DC    <ZH1SAZ+63*$AF
004675                                     *
004676                                     *
004677                                     *
004678  14EA  0000                ISATTL   RESV  $AF,0                   TSA LINK
004679  14EB  0000                ISATDV   RESV  1,0                     DEVICE AND LEVEL
004680  14EC  FFFF                ISATM1   DC    Z'FFFF'                 MASK 1
004681  14ED  0000                ISATM2   DC    Z'0000'                 MASK 2
004682  14EE  0000                ISATP    RESV  $AF,0                   P REG
004683  14EF  4000                ISATS    DC    X'4000'                 PRIV BIT
004684  14F0  0000                         RESV  9+7*$AF,0
004685                                     *
004686                                     *
004687  1500  0000                ISARTL   RESV  $AF,0                   TSA LINK
004688  1501  0000                ISARDV   RESV  1,0                     DEVICE AND LEVEL
004689  1502  FFFF                ISARM1   DC    Z'FFFF'                 MASK 1
004690  1503  0000                ISARM2   DC    Z'0000'                 MASK 2
004691  1504  0000                ISARP    RESV  $AF,0                   P REG
004692  1505  4000                ISARS    DC    X'4000'                 PRIV BIT
004693  1506  0000                         RESV  9+7*$AF,0
004694                                     *
004695                                     *
004696  1516  8A80 14EE           DEVIH    INC   <ISATP+$AF-1            RESET RETURN LOCATION
004697  1518  8A80 14EE                    INC   <ISATP+$AF-1
004698  151A  8A80 14E4                    INC   <DEVSEM                 SET SEMAPHORE
004699  151C  D800 14E6                    LDR   $R5,<DLVL               DEVICE LEVEL TO R5
004700  151E  E800 14E8                    LDR   $R6,<PLVL               PROGRAM LEVEL TO R6
004701  1520  D400 22BA                    OR    $R5,<CHAN               STICK IN CHANNEL NUMBER
004702  1522  A800 1501                    LDR   $R2,<ISARDV             LOOK AT RUPTING DEVICE
004703  1524  8852                        LBF   =$R2,=Z'0040'           SHUT OFF I/O BIT
        1525  0040
004704  1526  A955                        CMR   $R2,=$R5                WAS IT THE TAPE?
004705  1527  0906                        BE    >DEVIH1                 YES - OK
004706  1528  B800 0748                    LDR   $R3,<RPR3
004707  152A  D380 1030                    LNJ   $B5,<FNER               RUPTING DEVICE WAS NOT A TAPE
004708  152C  3339                FNER39   DC    '39'                    LABEL2
004709  152D  D800 14E6           DEVIH1   LDR   $R5,<DLVL               DEVICE LEVEL TO R5
004710  152F  8955                        LBT   =$R5,=Z'4000'           SET PRIVELEGE BIT
        1530  4000
004711  1531  8C52                        STS   =$R2                    STORE THE "S" REGISTER
004712  1532  A955                        CMR   $R2,=$R5                IS IT RIGHT?
004713  1533  0906                        BE    >DEVIH2                 YES - GO ON
004714  1534  B800 0748                    LDR   $R3,<RPR3
004715  1536  D380 1030                    LNJ   $B5,<FNER               'S' REG WRONG IN SAVE AREA
004716  1538  3430                FNER40   DC    '40'                    LABEL2
004717  1539  D570 003F           DEVIH2   AND   $R5,=Z'003F'            LOOK AT DEVICE LEVEL
004718  153B  8752                        CL    =$R2                    RESET INDEX
004719  153C  82A0 0000       X   DEVIH3   LB    <ZH1AFB,$R2             LOOK AT NEXT FLAG
004720  153E  A955                        CMR   $R2,=$R5                IS THIS FLAG THE DEVICE LEVEL?
004721  153F  090A                        BE    >DEVIH5                 YES - BRANCH
004722  1540  A956                        CMR   $R2,=$R6                IS THIS THE PROGRAM LEVEL FLAG
004723  1541  0906                        BE    >DEVIH5                 YES - BRANCH
004724  1542  0502                        BBT   >DEVIH4                 IF FLAG IS ON - BRANCH (SHOULDN'T BE)
004725  1543  0F91                        B     >DEVIH7                 OK FLAG IS OFF
```

```
004726  1544  B800 0748          DEVIH4 LDR   $R3,<KPR3
004727  1546  D380 1030                 LNJ   $B5,<FNER          ACT FLG IS ON WHEN IT SHOULDN'T BE
004728  1548  3431               FNER41 DC    '41'              LABEL2
004729  1549  0586               DEVIH5 BDF   >DEVIH6           FLAG IS OFF (SHOULDN'T BE)
004730  154A  A956                      CMR   $R2,=$R6          IS THIS THE PROGRAM LEVEL?
004731  154B  0909                      BE    >DEVIH7           YES - BRANCH
004732  154C  8820 0000       X                LDF   <ZHIAFB.$R2       NO - RESET DEVICE LEVEL FLAG
004733  154E  0F86                      B     >DEVIH7           CONTINUE
004734  154F  B800 0748          DEVIH6 LDR   $R3,<KPR3
004735  1551  D380 1030                 LNJ   $B5,<FNER          FLAG IS OFF, SHOULD BE ON
004736  1553  3432               FNER42 DC    '42'              LABEL2
004737  1554  A970 003F          DEVIH7 CMR   $R2,=63           IS THAT ALL THE FLAGS?
004738  1556  0282                      BGE   >DEVIH8           YES - BRANCH
004739  1557  27E5                      BINC  $R2,>DEVIH3       NO BUMP AND BRANCH
004740  1558  C870 8000          DEVIH8 LDR   $R4,=Z'8000'      GET READY FOR LEV **********
004741  155A  C400 14E7                 OR    $R4,<CLVL         *********
004742  155C  D900 14E7                 CMR   $R5,<CLVL         ARE LEVELS EQUAL?
004743  155E  0901 F1AB                 BE    RPK1              YES - DONT LEV
004744  1560  8E54                LEVXF  LEV   =$R4              GO BACK TO PROGRAM
004745  1561  0F81 FFB4                 B     DEVIH             BRANCH TO START OF HANDLER
004746                             *
004747                             *
004748                             *
004749                             *
004750  1563  B800 0748          ISATIH LDR   $R3,<KPR3
004751  1565  D380 1030                 LNJ   $B5,<FNER          CPU TRIED TO INTERRUPT
004752  1567  3433               FNER43 DC    '43'              LABEL2
004753  1568  0FFB                       B     >ISATIH           **********
004754                             ********************************************************************
004755                             *
004756                             *
004757  1569  DF80 1590          GOWRAP STB   $B5,<WRAPB5
004758  156B  D380 10B6                 LNJ   $B5,<SET           SET UP FOR RETRY MESSAGE
004759  156D  9B80 10C2                 LAB   $B1,<RT4           RETRY ROUTINE IS A NOP
004760  156F  9FC0 0C33                 STB   $B1,RTRN
004761  1571  9B80 1580                 LAB   $B1,<WRAPA         RETRY LOC
004762  1573  9F80 21A4                 STB   $B1,<RTRY
004763  1575  8756                      CL    =$R6               WRAP DATA OF ZEROS (0000)
004764  1576  D380 1591                 LNJ   $B5,<LDBUF
004765  1578  8700 218E                 CL    <OFSETW            WRITE OFFSET
004766  157A  8700 218D                 CL    <OFSETR            READ OFFSET
004767  157C  D380 1599                 LNJ   $B5,<TURN          DO TURN-AROUND TEST
004768  157E  D380 15F9                 LNJ   $B5,<COMPAR
004769                             *
004770  1580  D380 10B6          WRAPA  LNJ   $B5,<SET           SET UP FOR RETRY MESSAGE
004771  1582  9B80 158D                 LAB   $B1,<WRAPB         RETRY LOC
004772  1584  9F80 21A4                 STB   $B1,<RTRY
004773  1586  6CFF                      LDV   $R6,=-1            WRAP DATA OF ONES (FFFF)
004774  1587  D380 1591                 LNJ   $B5,<LDBUF
004775  1589  D380 1599                 LNJ   $B5,<TURN
004776  158B  D380 15F9                 LNJ   $B5,<COMPAR
004777  158D  DC80 1590          WRAPB  LDB   $B5,<WRAPB5
004778  158F  8385                      JMP   $B5
004779                             *
004780  1590  0000                WRAPB5 RESV  $AF,0
004781                             ********************************************************************
004782                             * FILL WBUF WITH DATA FROM R6
004783                             *
004784  1591  8751                LDBUF  CL    =$R1
004785  1592  9B80 17E6                 LAB   $B1,<WBUF
004786  1594  EF5D                LDBUFA STR   $R6,$B1,+$R1
004787  1595  1D09                      CMV   $R1,=9
004788  1596  027E                      BE    >LDBUFA
004789  1597  8751                      CL    =$R1
004790  1598  8385                      JMP   $B5
004791                             ********************************************************************
004792                             * WRAP DATA OUT
004793                             *
004794  1599  DF80 15F8          TURN   STB   $B5,<TURNB5
004795  159B  D380 175C                 LNJ   $B5,<FRB33         FILL READ BUFFER WITH 33'S
004796  159D  9800 218E                 LDR   $R1,<OFSETW        GET WRITE OFFSET
004797  159F  8190 17E6          TURNA  IOLD  <WBUF.$R1,<IOWRIT,=$R4
      15A1  0000 22BC
      15A3  0054
004798  15A4  07FB                      BIOF  >TURNA
004799  15A5  8000 22A7          TURNF  IO    <DIAGN,<OTCONF      DIAGNOSTIC CONFIGURATION
      15A7  0000 22BD
004800  15A9  07FC                      BIOF  >TURNF
004801  15AA  8000 22A4          TURNB  IO    <WRNOGO,<OTTASK     WRITE, NO MOTION
      15AC  0000 22C2
004802  15AE  07FC                      BIOF  >TURNB
004803  15AF  9800 218D                 LDR   $R1,<OFSETR        GET READ OFFSET
004804  15B1  9B80 1C66                 LAB   $B1,<KBUF          POINTER TO READ BUFFER
004805  15B3  8752                      CL    =$R2               RANGE COUNTER
004806  15B4  C800 22CA                 LDR   $R4,<INRANG        INPUT RANGE CHANNEL CONSTANT
004807  15B6  E800 22B3                 LDR   $R6,<TESTMD        TEST MODE CONTROL WORD
004808  15B8  F800 22BE                 LDR   $R7,<OTCONT        OUTPUT CONTROL WORD CHANNEL CONSTANT
004809  15BA  8055                TURNC  IO    =$R5,=$R4          (GET RANGE,)
      15BD  0054
004810  15BC  07FE                      BIOF  >TURNC             WAIT TILL DONE
004811                             *
004812                             * READ WRAPPED DATA BACK
004813                             *
004814  15BD  8056                TURND  IO    =$R6,=$R7          (SET TEST MODE)
      15BF  0057
004815  15BF  8030 22AC                 IO    <LCHNO,$R3,<OTCONT (OUTPUT LOGICAL CHAN NMBR)
      15C1  0000 22BE
004816  15C3  8000 22AA                 IO    <INDEX,<OTCONT     (SET INDEX MODE)
      15C5  0000 22BE
004817                             *
004818  15C7  8070 7C4E                 IO    =Z'7C4E',<OTCONT   (STOP THE TIMER)
      15C9  0000 22BE
004819  15CB  8070 88EA                 IO    =Z'88EA',<OTCONT   (SET SP TO TM#1)
      15CD  0000 22BE
004820  15CF  8070 2094                 IO    =Z'2094',<OTCONT   (CLEAR NON-DATA-SERVICE-REQUEST)
      15D1  0000 22BE
004821  15D3  8070 703E                 IO    =Z'703E',<OTCONT   (READ FIFO TO ACU)
      15D5  0000 22BE
004822  15D7  8070 A200                 IO    =Z'A200',<OTCONT   (WRITE FROM ACU TO SP)
      15D9  0000 22BE
004823  15DB  8070 2100                 IO    =Z'2100',<OTCONT   (DATA BYTE TAKEN, BUMP)
      15DD  0000 22BE
004824                             *
```

```
004825  15DF  8000 22B1            IO    <SCRDY,<OTCONT      (SET CHAN RDY)
        15E1  0000 22BE
004826  15E3  8000 22B4            IO    <WAITLP,<OTCONT     (BRANCH TO WAIT LOOP)
        15E5  0000 22BE
004827  15E7  8000 22B0            IO    <NOTEST,<OTCONT     (GET OUT OF TEST MODE)
        15E9  0000 22BE
004828                        *
004829  15EB  8055            TURNE IO    =$R5,<INWAIT        (XFER DATA TO R.H. BYTE OF R5)
        15EC  0000 22CF
004830  15EE  07FD                  BIOF  >TURNE
004831  15EF  D7DD                  STH   $R5,$B1.+$R1        PUT DATA IN BUFFER
004832  15F0  8AD2                  INC   =$R2                BUMP CNTR
004833  15F1  A900 219E             CMR   $R2,<RNG            COMPARE TO RANGE WRITTEN
004834  15F3  0200 15BD             BL    <TURND              B IF NOT DONE, GET MORE
004835                        *
004836  15F5  DC80 15F8             LDB   $B5,<TURNB5
004837  15F7  8385                  JMP   $B5
004838                        *
004839  15F8  0000            TURNB5 RESV  $AF,0
004840                        ********************************************************************
004841                        * COMPARE WRAPPED BACK DATA = DATA SENT
004842                        *
004843  15F9  DF80 160E       COMPAR STB   $B5,<CMPRB5
004844  15FB  D380 136E              LNJ   $B5,<GETRNG         INPUT RANGE REMAINDER
004845  15FD  D380 16EB              LNJ   $B5,<INIZ           INITIALIZE
004846  15FF  8980 219D              CMZ   <RANGE              CHECK IT
004847  1601  0904                  BE    >CMPRA              B IF OK, SB =0
004848  1602  D380 1030             LNJ   $B5,<FNER           RANGE REMAINDER SB = 0
004849  1604  3039            FNER09 DC    '09'                LABEL2
004850  1605  9B80 17E6       CMPRA LAB   $B1,<WBUF
004851  1607  AB80 1C66            LAB   $B2,<RBUF
004852  1609  D380 123F             LNJ   $B5,<ISB            COMPARE BYTES
004853  160B  DC80 160E             LDB   $B5,<CMPRB5
004854  160D  8385                  JMP   $B5
004855                        *
004856  160E  0000            CMPRB5 RESV  $AF,0
004857                        ********************************************************************
004858                        * FILL A BUFFER WITH BYTES OF INCREMENTING DATA.
004859                        *
004860                        *  $B1 = BUFFER
004861                        *  $R4 = RANGE
004862                        *  $R7 = STARTING DATA-BYTES
004863                        *
004864  160F  8F00 21A5       INCBFR SAVE  <SAVE1,=Z'7F7F'
        1611  7F7F
004865  1612  C800 219E             LDR   $R4,<RNG
004866  1614  C370 0002             DIV   $R4,=2              BYTES --> WORDS
004867  1616  FF71            INCBA  STR   $R7,+$B1
004868  1617  FA70 0202             ADD   $R7,=X'0202'
004869  1619  066A                  BCF   >INCBC
004870  161A  89D7                  CMZ   =$R7
004871  161B  0984                  BNE   >INCBB
004872  161C  F870 FF00             LDR   $R7,=Z'FF00'
004873  161E  0F85                  B     >INCBC
004874  161F  F970 0101       INCBB  CMR   $R7,=X'0101'
004875  1621  0982                  BNE   >INCBC
004876  1622  7C01                  LDV   $R7,=1
004877  1623  4773            INCBC  BDEC  $R4,>INCBA
004878  1624  8F80 21A5             RSTR  <SAVE1,=Z'7F7F'
        1626  7F7F
004879  1627  8385                  JMP   $B5
004880                        ********************************************************************
004881                        * DO A CARRIAGE RETURN/LINE FEED
004882                        *
004883  1628  DF80 1633       CRLF   STB   $B5,<CRLFB5
004884                               CALL  ZV$1.ZV$TC,NULL     (CRLF)
        162A  FBC0 0003
        162C  D380 0000        X
        162E  0F80
        162F  218C
004885  1630  DC80 1633             LDB   $B5,<CRLFB5
004886  1632  8385                  JMP   $B5                 RETURN
004887                        *
004888  1633  0000            CRLFB5 RESV  $AF,0
004889                        ********************************************************************
004890                        * CLEAR MODES TO -0 -1 -2 -3
004891                        *
004892  1634  9870 2D30       CLRMOD LDR   $R1,='-0'
004893  1636  2CFC                  LDV   $R2,=-4
004894  1637  9F20 2106       CLRMA  STR   $R1,<DXMD+4,$R2      STORE MODE
004895  1639  8AD1                  INC   =$R1
004896  163A  2780 1637             BINC  $R2,<CLRMA
004897  163C  8385                  JMP   $B5                 RETURN
004898                        ********************************************************************
004899                        * RUN QLT AND CHECK THE RESULTS.
004900                        *
004901  163D  DF80 16EA       QLTRUN STB   $B5,<QLTB5
004902  163F  9800 2186             LDR   $R1,<LABEL1         GET CURRENT LABEL1
004903  1641  9F00 21B8             STR   $R1,<TEMPB          SAVE IT
004904  1643  D380 0FA2             LNJ   $B5,<LABEL
004905  1645  514C            QL     DC    'QL'                LABEL1
004906  1646  8753                  CL    =$R3                DRIVE (PORT) =0
004907  1647  9C80 218C             LDB   $B1,<NULL
004908  1649  9F80 21B3             STB   $B1,<SLDB           CLEAR ERROR PARAMETERS
004909  164B  8700 21B4             CL    <STAT1
004910  164D  8700 21B5             CL    <STAT2
004911  164F  8700 2191             CL    <QLTFLG             WILL SET AFTER FIRST RUN OF QLT
004912  1651  8700 20F5             CL    <DRIVE              START WITH DRIVE ZERO
004913  1653  D380 1341             LNJ   $B5,<SETCHN         CONFIGURE CHANNEL CONSTANTS
004914                        *
004915                        * SET UP FOR MISSING RESOURCE TRAP
004916                        *
004917  1655  9B80 2311             LAB   $B1,<TSA1
004918  1657  9F80 2311             STB   $B1,<TSA1           WE DON'T WANT ANY TRAP FULL INTERRUPTS
004919  1659  9F80 0000       X     STB   $B1,<ZHNTSA
004920  165B  9B80 16E5             LAB   $B1,<QLTTH          TRAP HANDLER
004921  165D  9F80 0000       X     STB   $B1,<ZHTH15         TRAP VECTOR
004922                        *
004923                        * RUN QLT (ONE TIME ONLY)
004924                        *
004925  165F  8980 2191       QLTAA  CMZ   <QLTFLG
004926  1661  0980 1668             BNE   <QLTC               B IF QLT HAS ALREADY RUN
004927  1663  8000 22AB       QLTA   IO    <INZBDC,<OTCONT     RUN QLT, TRP TO QLTL VIA QLTTH IF NAK
```

```
                  1665  0000 22BE
004926            1667  07FC                    BIOF    >QLTA
004929                                  *
004930            1668  8900 2191       QLTC    LBT     <QLTFLG,=1              QLT HAS RUN, SET FLAG
                  166A  0001
004931            166B  8280 20E9               LB      <BDCFLG,=1              CHECK WHICH FIRMWARE LOAD
                  166D  0001
004932            166E  050B                    BBT     >QLTE                   B IF "BDC3"
004933                                  *
004934                                  * "BDC2" FIRMWARE, GET QLTI.
004935                                  *
004936            166F  3980 16CA               BNEZ    $R3,<QLTL               B IF NOT DRIVE ZERO, IE., DONE
004937            1671  8051           QLTD    IO      =$R1,<INQLT2            GET QLT
                  1672  0000 22C7
004938            1674  07FD                    BIOF    >QLTD
004939            1675  1048                    SOR     $R1,8                   RIGHT JUSTIFY
004940            1676  9F00 2192               STR     $R1,<QLTI               SAVE IT
004941            1678  0F89                    B       >QLTF                   CONTINUE
004942                                  *
004943                                  * "BDC3" FIRMWARE, GET QLTI
004944                                  *
004945            1679  8051           QLTE    IO      =$R1,<INQLT3            GET QLTI
                  167A  0000 22C8
004946            167C  07FD                    BIOF    >QLTE
004947            167D  9570 00FF               AND     $R1,=Z'00FF'            CLEAR LEFT BYTE
004948            167F  9F00 2192               STR     $R1,<QLTI               SAVE IT
004949                                  *
004950                                  * NOW CHECK QLT RESULTS. FIRST CHECK MDC.
004951                                  *
004952            1681  9570 000F       QLTF    AND     $R1,=Z'000F'            ($R1 STILL CONTAINS QLTI)
004953            1683  9970 000F               CMR     $R1,=Z'000F'
004954            1685  0906                    BE      >QLTG                   B IF MDC LOOKS OK
004955            1686  D380 1030               LNJ     $B5,<FNER               QLT FAILED, BUS OR MDC NG
004956            1688  5432           FNERT2  DC      'T2'                    LABEL2
004957            1689  0F80 16CA               B       <QLTL                   TRY NEXT CHANNEL
004958                                  *
004959                                  * NOW CHECK BITS FOR ADAPTER PORTS 0-3
004960                                  *
004961            168B  9800 2192       QLTG    LDR     $R1,<QLTI
004962            168D  1008                    SOL     $R1,8                   LEFT JUSTIFY
004963            168E  9F00 21B7               STR     $R1,<TEMPA              SAVE QLT FOR TESTING
004964            1690  8752                    CL      =$R2                    USE AS INDEX TO FOUR PORTS
004965            1691  8700 217E               CL      <FAIL                   WILL SET IF A PORT FAILED
004966            1693  9870 5030               LDR     $R1,=A'P0'              PORT ZERO LABEL2
004967            1695  9F00 169E               STR     $R1,<FNERP0             RESET LABEL2
004968                                  *
004969                                  * CHECK "QLTI" FOR ALL PORTS
004970                                  *
004971            1697  82A0 21B7       QLTH    LB      <TEMPA,$R2              GET QLT BIT
004972            1699  0506                    BBT     >QLTJ                   B IF PORT OK
004973            169A  8A80 217E               INC     <FAIL                   LOG THE FAILURE
004974            169C  D380 1030               LNJ     $B5,<FNER               ADAPTER PORT FAILED THE QLT
004975            169E  5030           FNERP0  DC      'P0'                    LABEL2
004976            169F  8AD2           QLTJ    INC     =$R2                    SET FOR NEXT PORT
004977            16A0  8A80 169E               INC     <FNERP0                 SET LABEL2
004978            16A2  2003                    CMV     $R2,=3
004979            16A3  03F4                    BLE     >QLTH                   B IF NOT DONE YET
004980            16A4  8980 217E               CMZ     <FAIL                   CHECK FOR FAILURES
004981            16A6  0900 16CA               BE      <QLTL                   B IF ALL PORTS OK
004982                                  *
004983                                  * NOW, IF "BDC3", REPORT RESULTS OF "QLTJ"
004984                                  *
004985            16A8  8280 20E9               LB      <BDCFLG,=1
                  16AA  0001
004986            16AB  0580 16CA               BBF     <QLTL                   B IF BDC2, DONE
004987            16AD  9800 2180               LDR     $R1,<FIRM               CHECK FIRMWARE REV
004988            16AF  1D21                    CMV     $R1,=X'21'              REV '21
004989            16B0  0200 16CA               BL      <QLTL                   B IF EARLIER THAN '21
004990            16B2  8051           QLTK    IO      =$R1,<INQLTJ            GET "QLTJ"
                  16B3  0000 22C9
004991            16B5  07FD                    BIOF    >QLTK
004992            16B6  1048                    SOR     $R1,8                   RIGHT JUSTIFY
004993            16B7  9970 00FF               CMR     $R1,=Z'00FF'
004994            16B9  0911                    BE      >QLTL                   B IF NO ERROR
004995            16BA  9F00 21B7               STR     $R1,<TEMPA              SAVE IT
004996                                          CALL    ZV$HA,TEMPA,VRBL        CONVERT CONTENTS TO ASCII
                  16BC  FBC0 0003
                  16BE  D380 0000       X
                  16C0  0F80
                  16C1  21B7
                  16C2  21BB
004997            16C3  9800 21BD               LDR     $R1,<VRBL+2             GET ASCII (LABEL2)
004998            16C5  9F00 16C9               STR     $R1,<FNERQL             SAVE AS LABEL2
004999            16C7  D380 1030               LNJ     $B5,<FNER               QLT FAILED, SEE HEADER FOR DETAILS
005000            16C9  0000           FNERQL  RESV    1,0                     SPACE FOR LABEL2
005001                                  *
005002                                  * SET UP FOR NEXT CHANNEL (DRIVE)
005003                                  *
005004            16CA  8AD3           QLTL    INC     =$R3
005005            16CB  9830 20F6               LDR     $R1,<DRIVE0,$R3         GET CHANNEL ADDRESS
005006            16CD  9F00 20F5               STR     $R1,<DRIVE
005007            16CF  D380 1341               LNJ     $B5,<SETCHN             CONFIGURE ADDRESSES
005008            16D1  3D03                    CMV     $R3,=3
005009            16D2  0360 165F               BLE     <QLTAA                  LOOP BACK IF NOT DONE YET
005010                                  *
005011                                  * TEST DONE, RESTORE TRAP HANDLER AND RETURN
005012                                  *
005013            16D4  8980 2191               CMZ     <QLTFLG
005014            16D6  0984                    BNE     >QLTM                   B IF QLT WAS RUN
005015            16D7  D380 1030               LNJ     $B5,<FNER               QLT NEVER WAS RUN, GOT NAK'ED
005016            16D9  5430           FNERT0  DC      'T0'                    LABEL2
005017            16DA  9B80 0F9A      QLTM    LAB     $B1,<TH15               NORMAL TRAP HANDLER FOR...
005018            16DC  9F80 0000       X              STB     $B1,<ZHTH15             ...MISSING RSOURCE TRAP
005019            16DE  9800 21B8               LDR     $R1,<TEMPB              GET ORIGINAL LABEL1
005020            16E0  9F00 2186               STR     $R1,<LABEL1             RESTORE IT
005021            16E2  DC80 16EA               LDB     $B5,<QLTB5              RETURN ADDRESS
005022            16E4  8385                    JMP     $B5                     RETURN
005023                                  *
005024                                  * TRAP HANDLER FOR MISSING RESOURCES IN TEST 'QLTRUN'.
005025                                  *
005026            16E5  9B80 16CA      QLTTH   LAB     $B1,<QLTL               CONTINUATION ADDRESS
005027            16E7  9F80 2317               STB     $B1,<TSAIP
005028            16E9  0003                    RTT                             RETURN
```

```
005029                                  *
005030                                  *
005031                                  *
005032  16EA  0000                      QLTB5  RESV   $AF,0
005033                                  ********************************************************************
005034                                  * DO BDC INITIALIZE.
005035                                  *
005036  16EB  0F80 1708                 INIZ   STB    $B5,<INIZB5
005037  16ED  8000 22AB                        IO     <INZBDC,<OTCONT        INITIALIZE AND RUN QLT
        16EF  0000 22BE
005038  16F1  07FA                             BIOF   >INIZ
005039  16F2  8000 21B7                 INIZA  IO     <TEMPA,<INRUPT         CHECK IF MDC IS ACCESSIBLE
        16F4  0000 22CB
005040  16F6  07FC                             BIOF   >INIZA
005041  16F7  9C80 218C                        LDB    $B1,<NULL
005042  16F9  9F80 21B3                        STB    $B1,<SLDB              CLEAR PARAMETERS FOR ERROR REPORT
005043  16FB  8700 21B4                        CL     <STAT1
005044  16FD  8700 21B5                        CL     <STAT2
005045                                  *
005046                                  * CHECK THAT INITIALIZE WORKED
005047                                  *
005048  16FF  8980 21B7                 INIZB  CMZ    <TEMPA                 CHECK RUPT CONTROL LEVEL, SB ZERO
005049  1701  0904                             BE     >INIZC                 B IF OK
005050  1702  0380 1030                        LNJ    $B5,<FNER              INIZ DIDN'T DO IT'S THING
005051  1704  3234                     FNER24 DC     '24'                   LABEL2
005052                                  *
005053  1705  0C80 1708                 INIZC  LDB    $B5,<INIZB5
005054  1707  8385                             JMP    $B5                    RETURN
005055                                  *
005056  1708  0000                      INIZB5 RESV   $AF,0
005057                                  ********************************************************************
005058                                  * CHECK MEASURED TIME AGAINST NOMINAL LIMITS, +/-35%.
005059                                  * NOMINAL VALUE IS FOR 45 IPS MACHINE.
005060                                  *
005061  1709  9CF5                      TIMCHK LDB    $B1,+$B5               POINTER TO NOMINAL VALUE
005062  170A  0F80 1744                        STB    $B5,<TIMCB5
005063                                  *
005064  170C  8051                     TIMCA  IO     =$R1,<INIDEN           GET ID CODE
        170D  0000 22C6
005065  170F  07FD                             BIOF   >TIMCA
005066  1710  9570 0003                        AND    $R1,=3                 SPEED BITS
005067  1712  1D01                             CMV    $R1,=1
005068  1713  0900 171B                        BE     <TIMCB                 B IF 45 IPS
005069  1715  1D02                             CMV    $R1,=2
005070  1716  0900 172F                        BE     <TIMCD                 B IF 75 IPS
005071  1718  0380 1030                        LNJ    $B5,<FNER              ID CODE INDICATES INVALID SPEED
005072  171A  3637                     FNER67 DC     '67'                   LABEL2
005073                                  *
005074                                  * CHECK TIMING (+/- 35 %) FOR 45 IPS DRIVE
005075                                  *
005076  171B  E801                      TIMCB  LDR    $R7,$B1                NOMINAL VALUE
005077  171C  EB70 0041                        MUL    $R7,=65                N X 65%
005078  171E  E370 0064                        DIV    $R7,=100
005079  1720  F900 21BA                        CMR    $R7,<TIMCNT
005080  1722  0309                             BG     >TIMCC                 B IF NG
005081  1723  EB70 0087                        MUL    $R7,=135               N X 135%
005082  1725  E370 0041                        DIV    $R7,=65
005083  1727  F900 21BA                        CMR    $R7,<TIMCNT
005084  1729  0300 1741                        BG     <TIMCF                 B IF OK
005085  172B  0380 1030                 TIMCC  LNJ    $B5,<FNER              TIME NG FOR 45 IPS DRIVE
005086  172D  3230                     FNER20 DC     '20'                   LABEL2
005087  172E  0F93                             B      >TIMCF                 RETURN
005088                                  *
005089                                  * CHECK TIMING (+/- 35%) FOR 75 IPS DRIVE
005090                                  *
005091  172F  E801                      TIMCD  LDR    $R7,$B1                GET NOMINAL VALUE
005092  1730  EB70 0027                        MUL    $R7,=39                N X 45/75 X 65%
005093  1732  E370 0064                        DIV    $R7,=100
005094  1734  F900 21BA                        CMR    $R7,<TIMCNT
005095  1736  0308                             BG     >TIMCE                 B IF TIME NG
005096  1737  EB70 0087                        MUL    $R7,=135               N X 45/75 X 135%
005097  1739  E370 0041                        DIV    $R7,=65
005098  173B  F900 21BA                        CMR    $R7,<TIMCNT
005099  173D  0304                             BG     >TIMCF                 RETURN
005100  173E  0380 1030                 TIMCE  LNJ    $B5,<FNER              TIME NG FOR 75 IPS DRIVE
005101  1740  3232                     FNER22 DC     '22'                   LABEL2
005102                                  *
005103                                  *
005104                                  *
005105  1741  0C80 1744                 TIMCF  LDB    $B5,<TIMCB5
005106  1743  8385                             JMP    $B5                    RETURN
005107                                  *
005108  1744  0000                      TIMCB5 RESV   $AF,0
005109                                  ********************************************************************
005110                                  * INVOKE THE PATCHING LIBRARY ROUTINE
005111                                  *
005112                                  PCH    CALL   ZV$PCH                 B <START RESTART
        1745  FBF0 0001
        1747  0380 0000          X
005113  1749  0F80 0237                        B      <NEXT                  RETURN TO NEXT
005114                                  ********************************************************************
005115                                  * REWIND TO BOT AND WAIT FOR RDY
005116                                  *
005117  174B  0F80 175B                 GOBOT  STB    $B5,<BOTB5
005118  174D  8730 20FE                        CL     <DXFN.$R3              CLEAR FILE COUNT
005119  174F  8730 210E                        CL     <DXRN.$R3              CLEAR RECORD COUNT
005120  1751  8000 2299                 GOBOTA IO     <REWIND,<OUTTASK
        1753  0000 22C2
005121  1755  07FC                             BIOF   >GOBOTA
005122  1756  0380 0FEC                        LNJ    $B5,<INSTAT            WAIT FOR BOT
005123  1758  0C80 175B                        LDB    $B5,<BOTB5
005124  175A  8385                             JMP    $B5                    RETURN
005125                                  *
005126  175B  0000                      BOTB5  RESV   $AF,0
005127                                  ********************************************************************
005128                                  * FILL READ BUFFER WITH '3333' FOR RANGE OF 2048+256 BYTES (1152 WORDS)
005129                                  *
005130  175C  0F80 1769                 FRB33  STB    $B5,<FRB3B5
005131                                         CALL   ZV$F,RBUF,X3333,LENGTH
        175E  FBC0 0003
        1760  0380 0000          X
        1762  0F80
        1763  1C66
```

```
         1764  21BF
         1765  2189
005132   1766  DC80 1769              LDB    $B5,<FRB3B5
005133   1768  8385                   JMP    $B5                    RETURN
005134                          *
005135   1769  0000              FRB3B5 RESV   $AF,0
005136                          ********************************************************************
005137                          * CHECK FOR E-O-T WHILE IN DEBUG MODE
005138                          *
005139   176A  DF80 177A          EOTCHK STB    $B5,<EOTB5
005140   176C  8280 21B4                 LB     <STAT1,=Z'0100'        EOT BIT
         176E  0100
005141   176F  0588                      BDF    >EOTRET                RETURN IF NOT EOT
005142                                   CALL   ZV$I.ZV$TC,MSEOT
         1770  FBC0 0003
         1772  0380 0000       X
         1774  0F80
         1775  2235
005143   1776  0000                      HLT                           HIT "EXECUTE" TO CONTINUE
005144                          *
005145   1777  DC80 177A          EOTRET LDB    $B5,<EOTB5
005146   1779  8385                   JMP    $B5                    RETURN
005147                          *
005148   177A  0000              EOTB5  RESV   $AF,0
005149                          ********************************************************************
```

```
005150                          * WAIT FOR N/BSY. FIRST MAKE SURE THAT SOME OPERATION HAS BEEN
005151                          * INITIATED ON THE DRIVE. THEN DO THE TIME-OUT. THREE TERMINATIONS
005152                          * ARE POSSIBLE:
005153                          *    1. NORMAL TERMINATION - NON INTERRUPT, WAIT FOR DEVICE N/BSY.
005154                          *    2. NORMAL TERMINATION - INTERRUPT.
005155                          *    3. TIMEOUT ERROR - RTC INTERRUPTS OUT OF WAIT LOOP. ISSUE A
005156                          *       "STOPIO" COMMAND AND CONTINUE.
005157                          *
005158  177B  0F80 17E5        WAIT    STB     $B5,<WAITB5
005159                          *
005160  177D  8000 219D                 IO      <RANGE,<INRANG          TRY AN IO INSTRUCTION
      177F  0000 22CA
005161  1781  0784                     BIOF    >WAITA                  B IF OK
005162  1782  D360 1030                LNJ     $B5,<FNER               SB IN MIDST OF OPERATION, SHOULDN'T...
005163  1784  3630              FNER60 DC      '60'                    LABEL   ...GET ACK'ED.
005164                          *
005165  1785  9870 0258        WAITA   LDR     $R1,=600                SET UP FOR 5 SEC TIMEOUT (AT 60 HZ)
005166  1787  9F00 0000      X         STR     $R1,<ZHRTCC             IF TIMEOUT, RUPT TO LEVV5
005167  1789  8700 0000      X         CL      <ZHRTCI
005168                          *
005169  178B  9B80 2300                 LAB     $B1,<SA5DV
005170  178D  9F80 0005      X          STB     $B1,<ZHISAZ+5*$AF       RTC INTERRUPT VECTOR
005171  178F  9B80 17C4                LAB     $B1,<LEVV5
005172  1791  9F80 2303                STB     $B1,<SA5P               RTC RUPT HANDLER
005173                          *
005174  1793  0004                     RTCN                            START THE CLOCK
005175                          *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
005176                          * START TO DO NORMAL WAIT TIMEOUT (RUNNING AT LEVEL 15)
005177                          *
005178  1794  8000 219D        WAITB   IO      <RANGE,<INRANG          WAIT FOR N/BSY
      1796  0000 22CA
005179  1798  07FC                     BIOF    >WAITB
005180  1799  0005                     RTCF                            DONE, STOP THE CLOCK
005181  179A  D380 0FC5                LNJ     $B5,<TIME               WAIT FOR RUPT (IF IT'S COMING)
005182  179C  0001                     DC      1                       1 MS
005183  179D  D380 0FEC                LNJ     $B5,<INSTAT             GET STAT
005184  179F  9080 2186                LDH     $R1,<LABEL1             CHECK LABEL FOR DEBUG MODE
005185  17A1  1D44                     CMV     $R1,='D'                IF DEBUG MODE, NO RUPTS ALLOWED
005186  17A2  0900 17E2                BE      <WAITD                  B IF MODE 'D'
005187  17A4  9800 2186                LDR     $R1,<LABEL1
005188  17A6  9970 4550                CMR     $R1,='EP'
005189  17A8  0900 17E2                BE      <WAITD                  B IF END-PASS, DON'T RUPT
005190  17AA  9830 20FE                LDR     $R1,<DXFN,$R3           GET FILE NMBR
005191  17AC  1D08                     CMV     $R1,=8                  GT 8 FILES?
005192  17AD  0A80 17E2                BALE    <WAITD                  B IF 8 OR LESS, SHOULDN'T RUPT
005193  17AF  D380 1030                LNJ     $B5,<FNER               RUPT EXPECTED BUT DIDN'T RUPT
005194  17B1  3631              FNER61 DC      '61'                    LABEL2   ...RUPT LVL=10, CP LVL=15
005195  17B2  0F9E                     B       >WAITE                  FLUSH PENDING RUPT, RETURN
005196                          *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
005197                          * WE RUPTED TO LEVV10. MAKE SURE THAT RUPT WAS EXPECTED (OVER 8 FILES).
005198                          *
005199  17B3  0005              LEVV10 RTCF                             STOP THE CLOCK
005200  17B4  9830 20FE                LDR     $R1,<DXFN,$R3           GET FILE NMBR
005201  17B6  1D08                     CMV     $R1,=8
005202  17B7  0306                     BG      >WAITG                  B IF RUPT EXPECTED, OK
005203  17B8  D380 0FEC                LNJ     $B5,<INSTAT
005204  17BA  D380 1030                LNJ     $B5,<FNER               SHOULDN'T HAVE RUPTED, DEV LVL SB 0
005205  17BC  3632              FNER62 DC      '62'                    LABEL2
005206                          *
005207  17BD  9B80 17E2        WAITG   LAB     $B1,<WAITD              RETURN ADR FROM WAIT ROUTINE
005208  17BF  9F80 230F                STB     $B1,<SA15P
005209  17C1  8E70 800F        LEVXH   LEV     =Z'8000'+15             SUSPEND IO LVL 15
005210  17C3  0FF0                     B       >LEVV10                 SET SA10P TO LVL 10 FOR NEXT TIME
005211                          *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
005212                          * WAIT TIMED OUT. STOP THE TRANSFER AND REPORT THE ERROR, THEN PICK UP
005213                          * THE PIECES AND RESUME
005214                          *
005215  17C4  0005              LEVV5   RTCF                            STOP THE CLOCK
005216  17C5  8000 22B2                 IO      <STOPIO,<OTCONT         STOP THE OPERATION IN IT'S TRACKS
      17C7  0000 22BE
005217  17C9  0704                     BIOT    >WAITC
005218  17CA  D380 1030                LNJ     $B5,<FNER               STOPIO SHOULD ALWAYS BE ACK'ED
005219  17CC  3633              FNER63 DC      '63'                    LABEL2
005220                          *
005221  17CD  D380 1030        WAITC   LNJ     $B5,<FNER               OPERATION TIMED OUT (5 SECONDS)
005222  17CF  3634              FNER64 DC      '64'                    LABEL2
005223                          *
005224  17D0  9C80 000A      X WAITE   LDB     $B1,<ZHISAZ+10*$AF
005225  17D2  8700 000A      X         CL      <ZHISAZ+10*$AF          INHIBIT LVL 10 RUPTS
005226  17D4  8C51                     STS     =$R1                    GET CURRENT CP LVL
005227  17D5  9570 003F                AND     $R1,=X'3F'              MASK IT
005228  17D7  1D0F                     CMV     $R1,=15
005229  17D8  0908                     BE      >WAITF                  B IF ALREADY AT LVL 15
005230  17D9  AB80 17E0                LAB     $B2,<WAITF
005231  17DB  AF80 230F                STB     $B2,<SA15P
005232  17DD  8E70 800F        LEVXI   LEV     =Z'8000'+15             SUSPEND TO LVL 15, BYPASS LVL 10
005233  17DF  0FE5                     B       >LEVV5                  RE-INSTATE SA5P TO LEVV5
005234                          *
005235  17E0  9F80 000A      X WAITF   STB     $B1,<ZHISAZ+10*$AF      RESTORE IV 10
005236                          *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
005237                          * WAIT IS DONE, RETURN TO PROGRAM
005238                          *
005239  17E2  DC80 17E5        WAITD   LDB     $B5,<WAITB5
005240  17E4  8385                     JMP     $B5                     RETURN
005241                          *
005242  17E5  0000              WAITB5 RESV    $AF,0
005243                          ********************************************************************
```

```
005244                          /*****************************************************************
005245                          *
005246                          * CONSTANTS, MESSAGES, TABLES, BUFFERS, ETC.
005247                          *
005248                          *****************************************************************
005249                          * CONSTANTS, BUFFERS
005250                          *
005251  17E6  0000              WBUF    RESV    (2048+256)/2,0          STANDARD WRITE BUFFER
005252  1C66  0000              RBUF    RESV    (2048+256)/2,0          STANDARD READ BUFFER
005253                          *
005254  20E6  0000              ABRN    RESV    $AF,0                   ABORT ROUTINE POINTER
005255  20E7  0000              AUFL    RESV    1,0                     =1 IF IN TEST AU, MODE A
005256  20E8  0000              BDC     RESV    1,0                     =BDC3 FOR BDC3 FIRMWARE, ===== FOR BDC2.
005257  20E9  0000              BDCFLG  RESV    1,0                     =1 FOR BDC3 FIRMWARE, =0 FOR BDC2 FIRMWARE.
005258  20EA  0000              BITE    RESV    1,0                     DRIVE NUMBER
005259  20EB  0000              CNT8    RESV    1,0                     COUNT 8 SPACE OPS FOR TEST AU
005260  20EC  0001              DC1     DC      1
005261  20ED  0002              DC2     DC      2
005262  20EE  1000              DC4096  DC      4096
005263  20EF  0003              DC3     DC      3
005264  20F0  0020              DC32    DC      32
005265  20F1  0008              DC8     DC      8                       BYTES X 2 FOR ZV$IA (4 BYTES)
005266  20F2  005C              DCBS    DC      2'005C'                 BACK-SLASH (NULL CHARACTER)
005267  20F3  0000              DECN    RESV    1,0                     UP TO 8 DATA ERRS BEFORE RETRY/ABORT
005268  20F4  0000              DEFL    RESV    1,0                     DATA ERR FLG, SET AFTER FRST ERR IN COMPARE
005269  20F5  0000              DRIVE   RESV    1,0                     CURRENTLY SELECTED UNIT (FROM "DRIVE0" ETC.)
005270  20F6  0000              DRIVE0  DC      2'0000'
005271  20F7  0080              DRIVE1  DC      2'0080'
005272  20F8  0100              DRIVE2  DC      2'0100'
005273  20F9  0180              DRIVE3  DC      2'0180'
005274  20FA  0000              DXDENS  RESV    4,0                     MASK, 1000 FOR PE, 0000 FOR NRZI
005275  20FE  0000              DXFN    RESV    4,0                     HEX FILE NUMBER FOR CURRENT DRIVE
005276  2102  0000              DXMD    RESV    4,0                     ASCII MODE (A,R,W) FOR CURRENT DRIVE
005277  2106  0000              DXPS    RESV    4,0                     DEC PASS COUNT FOR CURRENT DRIVE
005278  210A  0000              DXPT    RESV    4*$AF,0                 NEXT TEST FOR CURRENT DRIVE
005279  210E  0000              DXRN    RESV    4,0                     HEX RECORD NUMBER FOR CURRENT DRIVE
005280  2112  0000              EOTFLG  RESV    1,0                     =1 IF EOT DETECTED
005281  2113  0000              ERCT    RESV    1,0                     TOTAL ERRORS SINCE START OF MODE
005282  2114  0000              ERDRIV  RESV    1,0                     DRIVE NUMBER FOR "FNER"
005283  2115  0000              ERFILE  RESV    1,0                     FILE NUMBER FOR "FNER"
005284  2116  0064              ERLNTH  DC      (8+2*$AF)*10            TOTAL LENGTH OF ERROR TABLE
005285  2117  0000              ERREC   RESV    1,0                     RECORD NUMBER FOR "FNER"
005286  2118  0000              ERRFLG  RESV    1,0                     =1 AFTER FIRST REPORTED ERROR
005287  2119  0000              ERTB    RESV    (8+2*$AF)*10,0          SAVE FIRST TEN ERRORS FOR MODE "P"
005288  217D  2119              ERTBPT  DC      <ERTB                   RUNNING POINTER THROUGH ERTB
005289  217E  0000              FAIL    RESV    1,0                     =1 IF A "PORT" FAILED IN 'GLTRUN'
005290  217F  0000              FILES   RESV    1,0                     FILES PER PASS, A Q R W (0 = E-O-T)
005291  2180  0000              FIRM    RESV    1,0                     FIRMWARE REV '--XX'
005292  2181  0000              FLAG    RESV    1,0                     GENERAL PURPOSE FLAG
005293  2182  0000              GOFLAG  RESV    1,0                     IF SET, BYPASS "START"
005294  2183  0000              IDEN    RESV    1,0                     ID CODE
005295  2184  17E6              JJPT    DC      <WBUF                   WRITE BUFFER POINTER FOR DEBUG MODE
005296  2185  1C66              KKPT    DC      <RBUF                   READ BUFFER POINTER FOR DEBUG MODE
005297  2186  0000              LABEL1  RESV    1,0                     MAJOR ERROR REPORTING LABEL
005298  2187  0000              LABEL2  RESV    2,0                     MINOR ERROR REPORTING LABEL
005299  2189  0480              LENGTH  DC      (2048+256)/2            LENGTH (WORDS) OF WRITE/READ BUFFERS
005300  218A  0000              MXAD    RESV    $AF,0                   MAX ADDRESS WHERE RAND BUF MAY START
005301  218B  0000              NTST    RESV    $AF,0                   NEXT TEST AFTER AN ABORT
005302  218C  0000              NULL    RESV    $AF,0
005303  218D  0000              OFSETR  RESV    1,0                     TURN-AROUND TEST, READ OFFSET
005304  218E  0000              OFSETW  RESV    1,0                     TURN-AROUND TEST, WRITE OFFSET
005305  218F  0000              PFLAG   RESV    1,0                     MODE P PRINT FLAG
005306  2190  00FF              PTRN    RESV    1,X'FF'                 DATA PATTERN
005307  2191  0000              QLTFLG  RESV    1,0                     =1 IF QLT ALREADY RUN IN 'GLTRUN'
005308  2192  0000              QLTI    RESV    1,0                     "QLTI" FROM SCRATCHPAD (IN RIGHT BYTE)
005309  2193  0000              R3HOLD  RESV    1,0                     SAVE INDEX R3
005310  2194  0000              RANADR  RESV    $AF,0                   LAST ADDRESS FROM GRNADR
005311  2195  0000              RANDOM  RESV    8,0                     8 RAND NMBRS FOR TEST AU
005312  219D  0000              RANGE   RESV    1,0                     RANGE RCVD BY "GETRNG", ALSO IN T1-T7.
005313  219E  0800              RNG     RESV    1,X'800'                NUMBER OF BYTES, POSITIVE (DEFAULT = 2048)
005314  219F  0000              RNGFLG  RESV    1,0                     =1 TO SUPPRESS UNEQUAL LENGTH CHECKS
005315  21A0  003C              RTCHZ   DC      60                      DEFAULT RTC FREQUENCY
005316  21A1  0000              RTCN    RESV    1,0                     UP TO 2 MORE RETRIES BEFORE ABORT
005317  21A2  0000              RTFL    RESV    1,0                     RETRY FLAG (= 1 WHEN DOING RETRY)
005318  21A3  0000              RTRN    RESV    $AF,0                   POINTER TO RETRY ROUTINE
005319  21A4  0000              RTRY    RESV    $AF,0                   RETRY LOCATION
005320  21A5  0000              SAVE1   RESV    7+7*$AF,0               CONTEXT SAVE AREA
005321  21B3  0000              SLDB    RESV    $AF,0                   POINTER TO CURRENT DATA BUFFER
005322  21B4  0000              STAT1   RESV    1,0                     FIRST STATUS WORD
005323  21B5  0000              STAT2   RESV    1,0                     SECOND STATUS WORD
005324  21B6  0000              SUPRES  RESV    1,0                     ERROR SUPPRESSION FLAG (1= SUPPRESS REPORTS)
005325  21B7  0000              TEMP    RESV    3,0
005326        21B7              TEMPA   EQU     TEMP
005327        21B8              TEMPB   EQU     TEMP+1
005328        21B9              TEMPC   EQU     TEMP+2
005329  21BA  0000              TIMCNT  RESV    1,0                     COUNT MILLISECONDS FOR 'TIMCHK' ROUTINE
005330  21BB  2D2D 2D2D 2424    VBL     TEXT    '------$$'              VARIABLE CONVERSION BUFFER
        21BE
005331  21BF  3333              X3333   DC      2'3333'
005332  21C0  0000              ZVSTAT  RESV    1,0                     RETURNED STATUS FOR ZV$ ROUTINES.
005333                          *****************************************************************
005334                          * MESSAGES
005335                          *
005336        21C1              MSNAME  EQU     $
005341                          Z1      NULL
005342  21C1  4D54 5553 3320    MSG     TEXT    'MTU53 REV G, 9-TK PE/NRZI MAG TAPE TEST, 29 JUNE 78$'
        21C4  5245 5620 472C
              2039 2D54 4B20
              5045 2F4E 525A
              4920 4D41 4720
              5441 5045 2054
              4553 542C 2032
              3920 4A55 4E45
              2037 3824
005343                          Z2      NULL
005344  21DB  434F 5059 5249    CPYRIT  TEXT    '(C)COPYRIGHT 1976, 77, 78 BY HONEYWELL INFORMATION SYSTEMS, INC.'
        21DE  4748 5420 3139
              3736 2C20 3737
              2C20 3738 2042
              5920 484F 4E45
              5957 454C 4C20
              494E 464F 524D
              4154 494F 4E20
```

```
                5359 5354 454D
                532C 2049 4E43
                2E00
005345  21EA  0D0A                MSHEAD DC     X'0D0A'              CR/LF
005346  21EB  4552 5220 4C41           TEXT     'ERR LABEL   LOC    UNIT FILE  REC   BUF   STAT1 STAT2 '
        21FE  4245 4C20 2020
              4C4F 4320 2020
              2055 4E49 5420
              4649 4C45 2020
              5245 4320 2020
              4255 4620 2020
              5354 4154 3120
              5354 4154 3220
005347  2216  0D0A 2424                DC     2'0D0A2424'          CR/LF$$
005348                           *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
005349  2218  2041 424F 5254           MSABRT TEXT   ' ABORT$'
        221D  2400
005350  221C  2020 4259 5445           MSBYTE TEXT   '  BYTE  $'
        221F  2020 2400
005351  2221  4348 414E 4E45           MSCHAN TEXT   'CHANNEL$'
        2224  4C24
005352  2225  4452 4956 4520           MSDENS TEXT   'DRIVE "0" IS  --------$'
        2228  2230 2220 4953
              2020 2D2D 2D2D
              2D2D 2D2D 2400
005353  2231  4F46 4620 4C49           MSDOWN TEXT   'OFF LINE'
        2234  4E45
005354  2235  2045 2D4F 2D54           MSEOT  TEXT   ' E-O-T$'
        2238  2400
005355  2239  2045 5252 2853           MSERR  TEXT   ' ERR(S)   $'
        223C  2920 2020 2400
005356  223F  2048 4558 2046           MSFILE TEXT   ' HEX FILE(S)...$'
        2242  494C 4528 5329
              2E2E 2E24
005357  2247  4244 432D 2046           MSFIRM TEXT   'BDC- FIRMWARE REV$'
        224A  4952 4D57 4152
              4520 5245 5624
005358  2250  494E 5641 4C49           MSINVL TEXT   'INVALID MODE$'
        2253  4420 4D4F 4445
              2400
005359  2257  2C20 2049 5320           MSIS   TEXT   ',  IS  $'
        225A  2024
005360  225B  2020 4C49 4E4B           MSLINK TEXT   '  LINKS$'
        225E  5324
005361  225F  4D4F 4445 2024           MSMODE TEXT   'MODE $'
005362  2262  2020 4E45 5854           MSNEXT TEXT   '  NEXT$'
        2265  2400
005363  2266  4E52 5A49 2020           MSNRZI TEXT   'NRZI    '
        2269  2020
005364  226A  2020 5241 4E47           MSPARA TEXT   '  RANGE, DATA$'
        226D  452C 2044 4154
              4124
005365  2271  2020 5041 524D           MSPARM TEXT   '  PARMS:  $'
        2274  533A 2020 2400
005366  2277  2020 5041 5353           MSPASS TEXT   '  PASS$'
        227A  2400
005367  227B  5045 2020 2020           MSPE   TEXT   'PE      '
        227E  2020
005368  227F  514C 5420 5041           MSQLPS TEXT   'QLT PASS$$'
        2282  5353 2424
005369  2284  504F 5745 5220           MSRTC  TEXT   'POWER FREQ, HZ$'
        2287  4652 4551 2C20
              485A 2400
005370  228C  2052 4554 5259           MSRTRY TEXT   ' RETRY$'
        228F  2400
005371  2290  2020 5342 2020           MSSB   TEXT   '  SB  $'
        2293  2400
005372  2294  2020 5354 4154           MSSTAT TEXT   '  STAT:  $'
        2297  3A20 2024
005373                           ********************************************************************
005374                           * LIST OF TASK WORDS  (IO <REWIND,<OTTASK)
005375                           *
005376  2299  8000                REWIND DC     2'8000'              REWIND
005377  229A  C000                UNLOAD DC     2'C000'              UNLOAD
005378  229B  0800                FWDREC DC     2'0800'              FORWARD SPACE A RECORD (BLOCK)
005379  229C  0400                REVREC DC     2'0400'              BACKSPACE A RECORD (BLOCK)
005380  229D  1800                FWDFIL DC     2'1800'              FORWARD SPACE A FILE (TAPE MARK)
005381  229E  1400                REVFIL DC     2'1400'              BACKSPACE A FILE (TAPE MARK)
005382  229F  0900                READFD DC     2'0900'              READ FORWARD
005383  22A0  2800                ERASE  DC     2'2800'              ERASE
005384  22A1  3A00                WRITFM DC     2'3A00'              WRITE FILE (TAPE) MARK
005385  22A2  2B00                WRITE  DC     2'2B00'              WRITE
005386  22A3  0500                READRV DC     2'0500'              READ REVERSE (NOT IMPLEMENTED)
005387  22A4  2300                WRNOGO DC     2'2300'              WRITE WITH NO TAPE MOTION
005388  22A5  0000                NOPER  DC     2'0000'              NO OPERATION
005389                           ********************************************************************
005390                           * LIST OF CONFIGURATION WORDS (IO <NORMAL,<OTCONF)
005391                           *
005392  22A6  0800                ANSINH DC     2'0800'              INHIBIT ANSI MODE
005393  22A7  0A00                DIAGN  DC     2'0A00'              INHIBIT ANSI, DIAGNOSTIC MODE
005394  22A8  1A00                EVNPAR DC     2'1A00'              EVEN PARITY, ANSI INH, DIAG MODE
005395  22A9  0000                NORMAL DC     2'0000'              NORMAL MODE
005396                           ********************************************************************
005397                           * LIST OF CONTROL WORDS (IO <STOPIO,<OTCONT)
005398                           *
005399  22AA  A0C0                INDEX  DC     2'A0C0'              SET INDEX MODE
005400  22AB  6000                INZBDC DC     2'8000'              INITIALIZE BDC, DO QLT ON 4 PORTS OF BDC
005401  22AC  A030                LCHN0  DC     2'A030'              LOGICAL CHANNEL NMBR, PORT 0
005402  22AD  A034                LCHN1  DC     2'A034'              PORT 1
005403  22AE  A038                LCHN2  DC     2'A038'              PORT 2
005404  22AF  A03C                LCHN3  DC     2'A03C'              PORT 3
005405  22B0  0500                NOTEST DC     2'0500'              RESET TEST MODE
005406  22B1  4018                SCRDY  DC     2'4018'              SET CHANNEL RDY
005407  22B2  4000                STOPIO DC     2'4000'              RST BSY, RPT IF ENB, HLT DMA (AFFECTS 1 POI
005408  22B3  2000                TESTMD DC     2'2000'              SET TEST MODE
005409  22B4  0000                WAITLP RESV   1,0                  BRANCH TO WAIT LOOP INSTRUCTION
005410  22B5  9883                WRENB  DC     2'9883'              SET WRITE ENABLE, ETC
005411  22B6  8863                XFER1  DC     2'8863'              SET SCRATCH PAD MEM ADR TO STAT2
005412  22B7  BE00                XFER2  DC     2'BE00'              XFER STATUS TO SCRATCH PAD
005413  22B8  784E                WRINH  DC     2'784E'              RESET WRITE ORDER
005414  22B9  0008                CLADAP DC     2'0008'              ADAPTER HARD CLEAR
005415                           ********************************************************************
005416                           * CHANNEL CONSTANTS
```

```
005417                          *
005418  22BA  1400              CHAN    DC      Z'1400'         DEFAULT CHAN ADDRESS
005419                          *
005420  22BB  1409              IOREAD  DC      Z'1409'         IOLD READ
005421  22BC  1449              IOWRIT  DC      Z'1449'         IOLD WRITE
005422                          *
005423  22BD  1411              OTCONF  DC      Z'1411'         OUTPUT CONFIGURATION WORD
005424  22BE  1401              OTCONT  DC      Z'1401'         OUTPUT CONTROL WORD
005425  22BF  1403              OTRUPT  DC      Z'1403'         OUTPUT INTERRUPT CONTROL
005426  22C0  1419              OTSTW1  DC      Z'1419'         OUTPUT STATUS WORD 1
005427  22C1  141B              OTSTW2  DC      Z'141B'         OUTPUT STATUS WORD 2
005428  22C2  1407              OTTASK  DC      Z'1407'         OUTPUT TASK WORD
005429                          *
005430  22C3  1400              INBDC   DC      Z'1400'         INPUT BDC FIRMWARE LOAD TYPE
005431  22C4  1410              INCONF  DC      Z'1410'         INPUT CONFIGURATION WORD
005432  22C5  1404              INFIRM  DC      Z'1404'         INPUT FIRMWARE REV
005433  22C6  1426              INIDEN  DC      Z'1426'         INPUT ID CODE
005434  22C7  140A              INGLT2  DC      Z'140A'         INPUT BDC2 GLITCH TO LEFT BYTE
005435  22C8  1404              INGLT3  DC      Z'1404'         INPUT BDC3 GLITCH TO RIGHT BYTE
005436  22C9  140E              INGLTJ  DC      Z'140E'         INPUT BDC3 GLITCH TO LEFT BYTE
005437  22CA  140C              INRANG  DC      Z'140C'         INPUT RANGE
005438  22CB  1402              INRUPT  DC      Z'1402'         INPUT INTERRUPT CONTROL
005439  22CC  1418              INSTW1  DC      Z'1418'         INPUT STATUS WORD 1
005440  22CD  141A              INSTW2  DC      Z'141A'         INPUT STATUS WORD 2
005441  22CE  1406              INTASK  DC      Z'1406'         INPUT TASK WORD
005442  22CF  143E              INWAIT  DC      Z'143E'         INPUT WAIT LOOP LOCATION
005443                          *
005444        22D0              CHANZ   EQU     $               END OF CHANNEL WORDS
005445                          ********************************************************************
005446                          * TABLE OF DEBUG MODE LINKS
005447                          *
005448  22D0  0C3C              LINKS   DC      <MODED          FIRST TIME, START OVER AGAIN
005449  22D1  0000                      RESV    20*$AF,0
005450                          *- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
005451  22E5  0D0D              LNKTAB  DC      <AAX
005452  22E6  0D4A                      DC      <BB
005453  22E7  0D4F                      DC      <CC
005454  22E8  0D59                      DC      <DD
005455  22E9  0D72                      DC      <EE
005456  22EA  0D7D                      DC      <FFF
005457  22EB  0D88                      DC      <GG
005458  22EC  0C9B                      DC      <MODEDJ
005459  22ED  0D9B                      DC      <II
005460  22EE  0DA6                      DC      <JJ
005461  22EF  0DBD                      DC      <KK
005462  22F0  0DD4                      DC      <LL
005463  22F1  0E02                      DC      <MM
005464  22F2  0E0F                      DC      <NN
005465  22F3  0E2B                      DC      <OO
005466  22F4  0E34                      DC      <PP
005467  22F5  0E80                      DC      <QQ
005468  22F6  0E91                      DC      <RR
005469  22F7  0EC4                      DC      <SS
005470  22F8  0ECA                      DC      <TT
005471  22F9  0ED9                      DC      <UU
005472  22FA  0EF7                      DC      <VV
005473  22FB  0F02                      DC      <WW
005474  22FC  0F2B                      DC      <XX
005475  22FD  0F2D                      DC      <YY
005476  22FE  0F38                      DC      <ZZ
005477                          ********************************************************************
005478                          * LEVEL 5 INTERRUPT SAVE AREA (RTC LEVEL)
005479                          *
005480  22FF  0000              SA5TL   RESV    $AF,0           TSA LINK
005481  2300  0000              SA5DV   RESV    1,0             DEVICE
005482  2301  0000              SA5M1   DC      0               MASK 1
005483  2302  0000              SA5M2   DC      0               MASK 2
005484  2303  0000              SA5P    RESV    $AF,0           P REG
005485  2304  4000              SA5S    DC      Z'4000'         PRIV BIT
005486                          *
005487                          * LEVEL 10 INTERRUPT SAVE AREA (DEVICE INTERRUPT LVL, WHEN ENABLED)
005488                          *
005489  2305  0000              SA10TL  RESV    $AF,0           TSA LINK
005490  2306  0000              SA10DV  RESV    1,0             DEVICE
005491  2307  0000              SA10M1  DC      0               MASK 1
005492  2308  0000              SA10M2  DC      0               MASK 2
005493  2309  0000              SA10P   RESV    $AF,0           P REG
005494  230A  4000              SA10S   DC      Z'4000'         PRIV BIT
005495                          *
005496                          * LEVEL 15 INTERRUPT SAVE AREA (NORMAL RUNNING LEVEL)
005497                          *
005498  230B  0000              SA15TL  RESV    $AF,0           TSA LINK
005499  230C  0000              SA15DV  RESV    1,0             DEVICE
005500  230D  0000              SA15M1  DC      0               MASK 1
005501  230E  0000              SA15M2  DC      0               MASK 2
005502  230F  0000              SA15P   RESV    $AF,0           P REG
005503  2310  4000              SA15S   DC      Z'4000'         PRIV BIT
005504                          ********************************************************************
005505                          * TRAP SAVE AREA 1
005506                          *
005507  2311  0000              TSA1    RESV    $AF,0           TSA LINK
005508  2312  0000              TSA1I   RESV    1,0             I REGISTER
005509  2313  0000              TSA1R3  RESV    1,0             R3
005510  2314  0000              TSA1F   RESV    1,0             F
005511  2315  0000              TSA1Z   RESV    1,0             Z
005512  2316  0000              TSA1EA  RESV    $AF,0           EA
005513  2317  0000              TSA1P   RESV    $AF,0           P
005514  2318  0000              TSA1B3  RESV    $AF,0           B3
005515                          *
005516                          * TRAP SAVE AREA 2
005517                          *
005518  2319  0000              TSA2    RESV    $AF,0           TSA LINK
005519  231A  0000              TSA2I   RESV    1,0             I REGISTER
005520  231B  0000              TSA2R3  RESV    1,0             R3
005521  231C  0000              TSA2F   RESV    1,0             F
005522  231D  0000              TSA2Z   RESV    1,0             Z
005523  231E  0000              TSA2EA  RESV    $AF,0           EA
005524  231F  0000              TSA2P   RESV    $AF,0           P
005525  2320  0000              TSA2B3  RESV    $AF,0           B3
005526                          ********************************************************************
005527                          * TABLE OF NOMINAL TIMES FOR CHECKING TIMED OPERATIONS. VALUES
005528                          * IN MILLISECONDS ARE FOR A 45 IPS MACHINE. SUBROUTINE 'TIMCHK'
005529                          * WILL FORM LIMITS AT +/-35% OF THIS VALUE AND WILL TAKE 60% OF
```

```
005530                          * THESE NUMBERS TO GET THE NOMINAL VALUE FOR A 75 IPS MACHINE.
005531                          * IT WILL THEN FORM LIMITS AT +/-35% OF THIS NUMBER AND
005532                          * CHECK THE MEASURED VALUE AGAINST THE LIMITS.
005533                          *
005534  2321  00BB              TIMER   DC      187             TIME TO ERASE FROM BOT
005535  2322  0032              TIMFM   DC      50              TIME TO WRITE A FM FOLLOWING AN ERASE OP
005536                          ********************************************************************
005537                          * SEQUENCE OF TEST ROUTINES FOR MODES A Q R AND W
005538                          *
005539  2323  10C1              SEQN    DC      <RT2            NOP SO MARW DOESN'T BUMP DXPT TOO FAR
005540  2324  0400                      DC      <T1
005541  2325  04E9                      DC      <RW
005542  2326  0552                      DC      <ER
005543  2327  057F                      DC      <FM
005544  2328  05AC                      DC      <BR
005545  2329  05C5                      DC      <FR
005546  232A  05DE                      DC      <BF
005547  232B  05F7                      DC      <FF
005548  232C  0610                      DC      <EF
005549  232D  0687                      DC      <RP
005550  232E  074A                      DC      <SA
005551  232F  10C1              SEQ1    DC      <RT2
005552  2330  0A13                      DC      <FILE
005553  2331  0A37                      DC      <AAST
005554  2332  218C                      DC      <NULL           END OF SEQUENCE
005555                          ********************************************************************
005556                          * LIST OF TEST LOCATIONS FOR TEST "AU"
005557                          *
005558  2333  0A72              LIST    DC      <ABU
005559  2334  0A79                      DC      <ACU
005560  2335  0A80                      DC      <ADU
005561  2336  0A87                      DC      <AEU
005562  2337  0A8E                      DC      <AFU
005563  2338  0A95                      DC      <AGU
005564  2339  0A9C                      DC      <AHU
005565  233A  0AA3                      DC      <AIU
005566  233B  0AAA                      DC      <AJU
005567  233C  0AB1                      DC      <AKU
005568  233D  0AB8                      DC      <ALU
005569  233E  0ABF                      DC      <AMU
005570  233F  0AC6                      DC      <ANU
005571  2340  0AD1                      DC      <AOU
005572  2341  0AF8                      DC      <APU
005573  2342  0B2A                      DC      <ARU
005574                          ********************************************************************
005575  2343  0100                      END     MTUS3,START
0000  ERR COUNT
```

TITLE MTUS3,'REV G' NINE TRACK PE/NRZI MAG TAPE T&V (SAF)

```
      $AF       1005C   1014C   1369    1386    1387    1392    1397C   1408C   1409C   1432C
                1436    1511    2015C   2094C   2102    2599C   2819    2842    2943    2973
                3136    3137    3464    3467    3472    3475    3495C   3660C   3719    3772
                3798    3805    3864    3870    3873    3881    3918    3927    4014    4067
                4122    4170    4223    4310    4311    4312    4313    4357    4383    4414
                4446    4464    4475    4500    4510    4531    4547    4572    4582    4592
                4608    4664    4674    4678    4682    4684    4687    4691    4693    4696C
                4697C   4780    4839    4856    4888    5032    5056    5108    5126    5135
                5148    5170C   5224    5225C   5235C   5242    5254    5278    5284    5287
                5300    5301    5302    5310    5318    5319    5320    5321    5337    5339
                5449    5480    5484    5489    5493    5498    5502    5507    5512    5513
                5514    5518    5523    5524    5525
      $B1       1002    1003C   1004    1005C   1011    1012C   1013    1014C   1043    1045
                1046    1053    1054C   1055C   1133    1134C   1141    1142C   1174    1175C
                1176    1177C   1254    1257    1261    1264C   1279    1280    1289    1290C
                1315    1316B   1327    1328    1337    1338B   1345    1346B   1353    1354B
                1361    1362B   1368    1369    1369    1370    1373    1374C   1386    1408C
                1467    1470    1474    1477    1480    1483    1486    1489    1492    1495
                1502    1512    1589    1590C   1597    1613    1614C   1621    1632    1640
                1641C   1651    1659    1667    1668C   1682    1683C   1995    1996C   2014
                2015C   2019    2020C   2060    2061C   2080    2081C   2084    2085C   2091
                2092C   2093    2094C   2439    2440C   2442C   2446C   2450C   2451C   2484
                2485C   2543    2544C   2545    2598    2599C   2600    2601C   2759    2761
                2766    2767C   2768    2769C   2774    2774    2775C   2777C   2779C   2788
                2789C   2836    2837C   2926    2929    2972    2973    2973    2974C   3136
                3147C   3159C   3213    3225    3246    3261    3262C   3263C   3264    3265C
                3268    3285    3287C   3295    3298C   3306    3307C   3321    3322C   3324C
                3325    3326C   3337    3338C   3340C   3341    3342C   3493    3494C   3495C
                3496    3497C   3498    3499C   3566    3567C   3568C   3569    3570C   3571C
                3659    3660C   3676    3677C   3756    3757    3757    3758    3760    3763
                3763    3764    3768C   3819    3822C   3825C   3827C   3829C   3832C   3835C
                3838C   3841C   3844C   3847C   3849C   3939    3940C   3941C   3942    3943C
                3944    3945C   3946    3947C   3948    3949C   3971    3973C   3975C   4019
                4031C   4039C   4053C   4060C   4073    4074C   4075C   4076    4077C   4078
                4079C   4080    4081C   4082    4083C   4115    4228C   4282    4283    4303
                4304C   4387    4390    4391    4658    4659    4759    4760C   4761    4762C
                4771    4772C   4785    4786C   4804    4831C   4850    4867C   4907    4908C
                4917    4918C   4919C   4920    4921C   5017    5018C   5026    5027C   5041
                5042C   5061    5076    5091    5169    5170C   5171    5172C   5207    5208C
                5224    5235C
      $B2       1044    1046    1057    1058C   1387    1409C   1477    1478C   1511    1512
                1990    1992C   1993    2760    2761    2904    2905    3137    3158    3267
                3317    3318    3333    3334    3805    3806C   3827C   4116    4229C   4230
                4231    4263    4388    4391    4851    5230    5231C
      $B3       1388    1389C   1489    1490C   1991    1992C   2929    2930B   3660C   3837
                3838C
      $B4       1391    1393C   1394    1396    1397C   3146    3147C   3148B   3158    3159C
                3169    3170
      $B5       1016B   1027B   1040B   1108B   1123B   1137B   1168B   1172B   1186B   1189B
                1191B   1305    1306B   1313B   1314B   1325B   1334B   1339B   1342B   1347B
                1350B   1355B   1358B   1363B   1375B   1379C   1384B   1392    1394    1412B
                1428B   1433    1434B   1453B   1454B   1466B   1500B   1507    1508C   1514
                1515C   1516B   1541B   1543B   1546B   1550B   1553B   1557B   1560B   1564B
                1567B   1571B   1575B   1578B   1587B   1588B   1591B   1595B   1601B   1607B
                1612B   1615B   1619B   1625B   1631B   1636B   1639B   1642B   1646B   1652B
                1658B   1663B   1666B   1669B   1673B   1680B   1681B   1684B   1691B   1693B
                1701B   1704B   1707B   1711B   1718B   1723B   1735B   1755B   1762B   1768B
                1782B   1787B   1790B   1796B   1802B   1816B   1819B   1822B   1827B   1829B
                1830B   1838B   1841B   1846B   1848B   1849B   1857B   1860B   1865B   1867B
                1868B   1876B   1879B   1884B   1886B   1887B   1895B   1898B   1903B   1909B
                1911B   1914B   1920B   1921B   1926B   1931B   1934B   1938B   1950B   1957B
                1961B   1973B   1976B   1984B   2046B   2051B   2073B   2090B   2108B   2110B
                2118B   2120B   2128B   2131B   2139B   2144B   2147B   2155B   2162B   2165B
                2180B   2187B   2189B   2190B   2193B   2201B   2207B   2214B   2222B   2231B
                2233B   2237B   2246B   2248B   2249B   2257B   2258B   2264B   2265B   2273B
```

```
                2281B   2288B   2290B   2291B   2297B   2298B   2313B   2318B   2319B   2325B
                2326B   2339B   2346B   2348B   2354B   2362B   2368B   2374B   2384B   2392B
                2397B   2398B   2403B   2404B   2412B   2418B   2421B   2422B   2430B   2435B
                2455B   2465B   2466B   2474B   2479B   2490B   2491B   2498B   2503B   2517B
                2531B   2534B   2539B   2541B   2549B   2558B   2561B   2562B   2571B   2574B
                2575B   2587B   2590B   2595B   2614B   2625B   2631B   2633B   2635B   2645B
                2647B   2653B   2655B   2661B   2663B   2669B   2671B   2677B   2679B   2685B
                2687B   2693B   2695B   2701B   2703B   2709B   2711B   2717B   2719B   2725B
                2727B   2733B   2735B   2741B   2743B   2746B   2750B   2752B   2755B   2776B
                2778B   2783B   2791B   2802B   2808B   2813B   2816B   2817B   2827B   2832B
                2839B   2850B   2856B   2860B   2862B   2863B   2873B   2877B   2879B   2883B
                2886B   2891B   2893B   2894B   2895B   2913B   2921B   2934C   2935B   2941
                2942B   2949B   2955B   2961B   2987B   2992B   3000B   3011B   3023B   3030B
                3036B   3056B   3109B   3222B   3234B   3236B   3237B   3253B   3254B   3259B
                3266B   3269B   3270B   3275B   3277B   3278B   3279B   3323B   3327B   3339B
                3343B   3385B   3398B   3400B   3401B   3402B   3403B   3426B   3431B   3433B
                3434B   3451B   3457B   3463B   3471B   3477B   3486B   3491B   3503B   3509B
                3511B   3512B   3513B   3518B   3519B   3524B   3526B   3529B   3530B   3535B
                3539B   3542B   3546B   3550B   3555B   3557B   3558B   3559B   3564B   3573B
                3579B   3580B   3581B   3582B   3591B   3593B   3594B   3595B   3600B   3613B
                3616B   3623B   3628B   3633B   3636B   3642B   3661B   3667    3669B   3698B
                3704    3711C   3714    3715B   3741B   3749B   3755C   3769    3770B   3776C
                3779    3787    3792    3795    3796B   3802    3804C   3805    3858C   3862B
                3869B   3877B   3878    3879B   3889B   3900B   3908B   3910C   3911B   3912B
                3913B   3914B   3915    3916B   3920C   3921B   3922B   3923B   3924    3925B
                3931B   3935    3937    3939    3942    3944    3946    3948    3951C   3955B
                3958B   3961B   3977B   3995B   3996B   4007B   4010B   4011    4012B   4018C
                4064    4065B   4071    4073    4076    4078    4080    4082    4085C   4089B
                4092B   4095B   4109B   4110B   4111B   4117B   4119    4120B   4126    4128
                4131C   4133B   4144B   4150B   4154B   4155B   4165B   4167    4168B   4174C
                4190B   4201B   4207B   4211B   4212B   4220    4221B   4227C   4241B   4255B
                4266B   4271    4272B   4290C   4305B   4306B   4319B   4328B   4329    4330B
                4334C   4337B   4346B   4351B   4353B   4354    4355B   4361C   4370B   4377B
                4379B   4380    4381B   4393B   4397C   4399B   4400B   4409B   4411    4412B
                4421B   4425C   4426B   4427B   4428B   4443B   4445    4446B   4452C   4453B
                4461    4462B   4468C   4472    4473B   4479C   4480B   4481B   4482B   4491B
                4497    4498B   4504C   4507    4508B   4514C   4515B   4516B   4517B   4526B
                4528    4529B   4535C   4544    4545B   4551C   4552B   4553B   4554B   4563B
                4569    4570B   4576C   4579    4580B   4586C   4588B   4589    4590B   4596C
                4599B   4603B   4605    4606B   4612C   4642B   4647    4648B   4652B   4707B
                4715B   4727B   4735B   4751B   4757C   4758B   4764B   4767B   4768B   4770B
                4774B   4775B   4776B   4777    4778B   4790B   4794C   4795B   4836    4837B
                4843C   4844B   4845B   4848B   4852B   4853    4854B   4879B   4883C   4885
                4886B   4897B   4901C   4904B   4913B   4955B   4974B   4999B   5007B   5015B
                5021    5022B   5036C   5050B   5053    5054B   5061    5062C   5071B   5085B
                5100B   5105    5106B   5117C   5122B   5123    5124B   5130C   5132    5133B
                5139C   5145    5146B   5158C   5162B   5181B   5183B   5193B   5203B   5204B
                5218B   5221B   5239    5240B
$B6             1338B   1346B   1354B   1362B   1687B   1758B   1791B   1823B   1842B   1861B
                1880B   1899B   1977B   1982C   2098    2099B   2563B   2591B   2975B   3037B
                3039B   3040B   3045B   3170    3171B   3906B
$B7             2543    3316    3318    3321    3332    3334    3337    4291    4292C
$R1              994     995    1020    1021    1022    1023C   1024C   1042    1045    1110
                1111C   1147    1148    1153    1154C   1160    1161C   1184    1185C   1188
                1190    1197    1198B   1216    1217    1219    1221    1223    1225    1227
                1229    1231    1233    1235    1237    1239    1241    1243    1245    1247
                1266    1267    1268C   1270    1271    1272    1280    1281    1293    1294C
                1297    1298C   1328    1329    1380    1381    1382    1399    1400    1401
                1410    1411C   1413    1414    1415C   1426    1427    1451    1452C   1470
                1471B   1472C   1474    1475C   1480    1481C   1483    1484C   1486    1487C
                1492    1493C   1495    1496C   1502    1503C   1580    1581C   1582    1583
                1602    1603C   1604C   1626    1627C   1654    1655C   1714    1715    1716
                1717B   1728    1729    1730B   1731    1732    1733    1734B   1746    1747
                1751    1752    1753    1775    1776    1778    1779    1780    1809    1810
                1812    1813    1814    1831    1832    1834    1835    1836    1850    1851
                1853    1854    1855    1869    1870    1872    1873    1874    1888    1889
                1891    1892    1893    1922    1923    1924    1943    1944    1946    1947
                1948    1966    1967    1969    1970    1971    1988    1989    2005    2006
                2007    2008    2009C   2010    2011    2012    2013B   2017    2018C   2020C
                2039    2040B   2041    2042B   2057    2058C   2059    2062    2083    2085C
                2132    2133    2135    2136    2137    2148    2149    2151    2152    2153
                2166    2172    2174    2176    2177    2178    2194    2195    2197    2198
                2199    2208C   2209    2215    2216    2218    2219    2220    2259C   2260
                2266    2267    2269    2270    2271    2276    2277    2292    2293    2302C
                2303    2306    2307    2309    2310    2311    2320    2321    2329    2332
                2333    2335    2336    2337    2349    2350    2355    2356    2358    2359
                2360    2378    2379C   2385    2386    2388    2389    2390    2405    2406
                2408    2409    2410    2423    2424    2426    2427    2428    2438    2442C
                2450C   2451C   2458    2459C   2467    2468    2470    2471    2472    2482
                2483C   2486    2492    2493    2495    2496    2550    2551    2553    2554
                2555    2556    2576    2577    2579    2580    2584    2585    2603    2604
                2902    2903C   2927    2928C   2929    2960    2961    2967    2968C   2969
                2999    3005    3006C   3017    3018C   3019C   3042    3043    3044C   3059
                3060    3061C   3068    3069    3074    3107    3108C   3135    3143    3145B
                3147C   3157B   3159C   3214C   3218C   3220C   3221    3223    3225    3231
                3247C   3284C   3287C   3288    3294C   3298C   3299    3305C   3307C   3307C
                3308C   3309    3408    3409C   3414    3415    3416    3417    3419    3420C
                3424    3425C   3443    3444    3445C   3482    3483    3484C   3543    3544
                3686    3688    3697B   3704    3705C   3712C   3729    3731    3732    3735
                3736    3783    3784    3802    3803C   3809    3810C   3811    3812C   3813
                3814    3821    3822C   3824    3825C   3831    3832C   3834    3835C   3840
                3841C   3843    3844C   3846    3847C   3887    3888C   3894    3895C   3896
                3897C   3935    3936C   3937    3938C   3962    3963    3965    3966    3972
                3973C   3974    3975C   4000    4001    4003    4004    4005    4020    4021
                4023    4025    4031C   4039C   4040C   4047    4048    4049C   4052    4053C
                4059    4060C   4071    4072C   4126    4127C   4128    4129C   4130C   4175
                4176    4177C   4179    4180    4181C   4182    4184    4185C   4186C   4230
                4231    4233    4235    4236    4238    4239    4263    4264    4277    4278
                4293    4294C   4345    4350    4362    4363    4365    4366    4389    4390
                4401    4402    4403    4405    4406    4407    4429    4430    4431    4433
                4434    4440    4441    4456    4457    4483    4484    4485    4487    4488
                4489    4518    4519    4520    4522    4523    4524    4538    4539    4555
                4556    4557    4559    4560    4561    4587    4613    4614    4622    4623
                4625    4627    4629    4631    4633    4635    4636    4653    4654    4784C
                4786C   4787    4789C   4796    4797    4803    4831C   4892    4894C   4895C
                4902    4903C   4937    4939    4940C   4945    4947    4948C   4952    4953
                4961    4962    4963C   4966    4967C   4987    4988    4990    4992    4993
                4995C   4997    4998C   5005    5006C   5019    5020C   5064    5066    5067
                5069    5165    5166C   5184    5185    5187    5188    5190    5191    5200
                5201    5226    5227    5228
$R2             1025    1026C   1061    1062    1063    1064    1065    1066    1078    1080
                1081C   1187    1188    1192B   1206    1207    1326    1331B   2068    2069
                2070    2121    2122    2124    2125    2126    2169    2170    2299    2300
```

```
                2303    2304    2329    2330    2437    2442C   2443    2446C   2447    2581
                2582    3140    3141    3151    3152    3154    3156    3158    3215    3233B
                3286    3287C   3297    3298C   3690    3691B   3730C   3734C   3737C   3740C
                3741B   4030C   4031C   4032    4038C   4039C   4041    4050C   4053C   4054
                4060C   4061    4250C   4251    4252    4257C   4258    4263    4340C   4435
                4436    4438    4702    4703    4704    4711    4712    4718C   4719    4720
                4722    4730    4732    4737    4739B   4805C   4832C   4833    4893    4894C
                4896B   4964C   4971    4976C   4978
        $R3     1060C   1061    1087    1111C   1125C   1127C   1128    1207    1208    1210
                1212    1214    1215C   1260    1261    1263    1264C   1267    1268C   1278
                1279    1283B   1292    1294C   1295C   1296C   1298C   1302    1315    1333
                1337    1341    1345    1349    1353    1357    1361    1368    1374C   1380
                1410    1414    1427    1715    1732    1752    1779    1813    1835    1854
                1873    1892    1947    1970    1983C   2045    2050    2072    2089    2125
                2136    2152    2177    2198    2219    2234C   2235C   2270    2310    2336
                2359    2372    2389    2409    2427    2471    2509    2555    2580    2581
                2603    2908    2927    2960    2967    2972    2974C   2999    3005    3017
                3033C   3034C   3035C   3043    3044C   3058    3060    3061C   3107    3504C
                3548C   3549C   3574C   3605C   3610    3611    3729    3777C   3808C   3809
                3811    3829C   3831    3834    3846    3972    3974    3987C   4004    4047
                4048    4101C   4175    4176    4305    4374C   4375C   4406    4434    4435
                4456    4469C   4470C   4488    4495C   4496C   4505C   4523    4536C   4537C
                4538    4560    4567C   4568C   4577C   4635    4706    4714    4726    4734
                4750    4815    4906C   4936B   5004C   5005    5008    5118C   5119C   5190
                5200
        $R4     1548    1549C   1555    1556C   1562    1563C   1569    1570C   1598    1599C
                1605    1606C   1610    1611C   1622    1623C   1629    1630C   1633    1634C
                1648    1649C   1656    1657C   1660    1661C   1675    1676C   2077    2078
                2079    3216    3217C   3225    3226C   3228    3229C   3674    3675C   3684C
                3692B   3696    3706    4740    4741    4744    4797    4806    4809    4865
                4866    4877B
        $R5     1067C   1068    1070C   1071    1072    2021    2026B   2027    2028    2030
                2031C   2038    2039    2053B   2054    2066    2070    2075    2086B   2905
                2906    2907C   2908    2909B   2914B   2919C   2920B   2922B   3667    3668C
                4699    4701    4704    4709    4710    4712    4717    4720    4742    4809
                4829    4831C
        $R6     1577    1579    1581C   1608    1609C   1997    2006    2011    2017    2032C
                2033C   2041    2056C   2082    2083    2087B   3117    3122B   3123    3125B
                3126    3128C   3610    4251    4253    4700    4722    4730    4763C   4773
                4786C   4807    4814
        $R7     1600    1624    1635    1650    1662    1664    1665C   2034    2036B   2064
                2065B   3118    3129C   3220C   3223    3611    4252    4253    4808    4814
                4867C   4868    4870    4872    4874    4876    5076    5077    5078    5079
                5081    5082    5083    5091    5092    5093    5094    5096    5097    5098
2596    AA
2607    AA1     2608B
2611    AA2     2605B   2612B
2614    AA3     2609B   2619
2633    AA4     2626B   2630B
2635    AA5     2639
3215    AAA     3235B   3248B
3225    AAB     3233B
3236    AAC     3232B
3240    AAD     3229C   3230
2595    AAST    5553
3213    AAX     5451
2645    AB      2621    2627B   2641
3931    ABR3    2243    2254    2620    2640    2797    2822    2845    2868    4139    4160
                4196    4217
5254    ABRN    3571C   3947C   4081C   4306B
2647    ABU     5558
2653    AC
2655    ACU     5559
2661    AD
2663    ADU     5560
2669    AE
2671    AEU     5561
2677    AF
2679    AFU     5562
2685    AG
2687    AGU     5563
2693    AH
2695    AHU     5564
2701    AI
2703    AIU     5565
2709    AJ
2711    AJU     5566
2717    AK
2719    AKU     5567
2725    AL
2727    ALU     5568
2733    AM
2735    AMU     5569
2741    AN
4596    ANB     3573B
4601    ANBA    4598B
4605    ANBB    4602B
4608    ANBB5   4596C   4605
5392    ANSINH  2375    2419
2743    ANU     5570
2750    AO      2745B
2752    AOU     5571
2783    AP      2770B
2791    APA     2786B   2796
2759    APWRS   2754B
2774    APWRSA  2762B
2785    APU     5572
2813    AQ      2803B
2816    AQA     2807B
2817    AQB     2821
2819    AQC     2767C   2777C   2789C
2825    AQD     2823
2832    AR      2798    2804B   2826B
2839    ARA     2835B   2844
2842    ARC     2769C   2779C   2837C
2834    ARU     5573
2860    AS      2851B
2862    ASA     2855B
2863    ASB     2867
2871    ASC     2846    2852B   2869
2877    AT      2872B
2886    ATA     2880B   2881B
2891    AU      2884B
```

```
2905  AU1       2947B
2913  AU2       2914B
2919  AU3       2909B
2921  AU4       2922B
2926  AU5       2915B   2920B
2934  AU6       2746B   2755B   2827B   2873B   4165B
2941  AU6A      2937B
2943  AU6B5     2934C   2941
2945  AU6C      2939B
2953  AU7       2867B
2965  AU8       2958B
2972  AU9       2963B   2966B
5255  AUFL      1179C   2629    2744    2753    2785    2806    2825    2834    2854    2871
                2897    2948C   3981    4148    4163    4205
3246  BB        5452
5256  BDC       1076    1147
1160  BDC3      1149B
5257  BDCFLG    1155C   1162    1537    2167    2370    2505    3079    4931    4985
1076  BDC10     1077B
1865  BF        5546
1876  BFA       1871B
1879  BFB       1875B
5258  BITE      1215C   1260    1263    1270    1292    3058
4592  BLANB5    4586C   4589
4586  BLANKS    3011B   3451B   3457B   3463B   3471B   3613B   3618B   3623B   3628B   3633B
                3869B
5126  BOTB5     5117C   5123
1827  BR        5544
1838  BRA       1833B
1841  BRB       1837B
4535  BSF       1867B   4515B
4541  BSFA      4542B
4543  BSFB      4540B
4547  BSFB5     4535C   4544
4514  BSFW      2893B   2894B   3433B
4526  BSFWA     4521B
4531  BSFWB5    4514C   4528
4528  BSFWC     4525B
4576  BSR       1829B   4552B
4582  BSRB5     4576C   4579
4551  BSRW      1911B   1921B   2631B   2808B   2856B   2913B   3529B   3911B   3912B   3921B
                3922B   4150B   4207B
4563  BSRWA     4558B
4572  BSRWB5    4551C   4569
4565  BSRWC     4562B
4568  BSRWD
4569  BSRWE     4566B
3252  CC        5453
5418  CHAN      1039    1042    4387    4701
5444  CHAN2     1044    4388
5414  CLADAP    2525
4894  CLRMA     4896B
4892  CLRMOD    1172B   1466B
4672  CLVL      2033C   2056C   2075    2078    2082    4741    4742
1263  CMNA      1255B   1258B
1292  CMNB      1282B
1278  CMNC      1273B
1279  CMNL      1283B
4850  CMPRA     4847B
4856  CMPRB5    4843C   4853
5259  CNTB      2903C   2945C   2946
4843  COMPAR    1591B   1615B   1642B   1669B   1684B   4768B   4776B
1379  COMX      1334B   1342B   1350B   1358B
1386  COMXA     1383B
1432  COMXB     1419B
1436  COMXB5    1379C   1432C   1433
1408  COMXC     1388    1402B
1433  COMXD     1417B   1421B   1430B
1393  COMXF     1395B
4670  CPCHAN    1989    2027
5344  CPYRIT
4883  CRLF      1186B   3023B   3236B   3642B
4888  CRLFB5    4883C   4885
3056  D1
4255  DATANG    4291
5260  DC1       3296    4051
5261  DC2       2380    3120
5263  DC3       3138
5264  DC32      1905    2545    4178
5262  DC4096    2458    2460
5265  DC8       1196    1677    2898
5266  DCBS      3465    3467    3473    3475    3671    3873
3259  DD        5454
3270  DDA       3261
3276  DE
5267  DECN      3888C   3895C   4322C   4323
5268  DEFL      3885C   3898C   4269    4317
4317  DERA      4296B   4299B
4322  DERB      4318B
4328  DERC      4320B
4696  DEVIH     1995    2080    4745B
4709  DEVIH1    4705B
4717  DEVIH2    4713B
4719  DEVIH3    4739B
4726  DEVIH4    4724B
4729  DEVIH5    4721B   4723B
4734  DEVIH6    4729B
4737  DEVIH7    4725B   4731B   4733B
4740  DEVIH8    4738B
4669  DEVSEM    2025C   2043    2048    4698C
5393  DIAGN     4799
4671  DLVL      2031C   2066    4699    4709
3399  DM
3432  DO
3492  DR
5269  DRIVE     1272    1411C   3068    3108C   3443    4389    4912C   5006C
5270  DRIVEU    1061    1410    3107    5005
5271  DRIVE1
5272  DRIVE2
5273  DRIVE3
3525  DT
3536  DU
3556  DV
```

```
3565  DW
5274  DXDENS  1111C  1125C  1715   1732   1752   1779   1813   1835   1854   1873
              1892   1947   1970   2125   2136   2152   2177   2198   2219   2270
              2310   2336   2359   2372   2389   2409   2427   2471   2555   2580
              4004   4406   4434   4488   4523   4560
5275  DXFN    1295C  2234C  2603   2960   2967   3017   3033C  3548C  3610   3809
              3831   3972   4047   4175   4365   4374C  4456   4469C  4495C  4536C
              4538   4567C  5118C  5190   5200
5276  DXMD    1188   1268C  1279   1294C  1327   1380   1427   2581   2999   3044C
              30616  3729   4435   4635   4894C
5277  DXPS    1298C  1450C  1452C  1455C  1456C  1457   1461   3005   3035C  3846
5278  DXPT    1261   1264C  1337   1345   1353   1361   1368   1374C  2972   2974C
5279  DXRN    1296C  2235C  2908   2927   3034C  3504C  3549C  3574C  3611   3811
              3834   3974   3987C  4048   4101C  4176   4375C  4470C  4496C  4505C
              4537C  4568C  4577C  5119C
3592  DY
3275  EE      5455
1903  EF      5548
1920  EFA     1913B
1907  EFB     1908B
1931  EFC     1925B  1933B
1942  EFD     1937B
1950  EFE     1945B
1955  EFF     1940B  1949B  1956B
1965  EFG     1960B
1973  EFH     1968B
1976  EFI     1916B  1963B  1972B
1905  EFJ     1906B
5148  EOTBS   5139C  5145
5139  EOTCHK  3279B  3403B  3513B  3559B  3582B  3595B
5280  EOTFLG  1178C  2953   3024C  3164   3999
5145  EOTRET  5141B
2982  EP
2998  EP1     2988B
3033  EP2     3029B
3042  EP3     3038B
2984  EP4     2985B
1762  ER      1765B  5542
1768  ERA     1772B
5383  ERASE   1544   1764   4398
4397  ERAW    1920B  2110B  2190B  3277B  3914B
4409  ERAWB   4404B
4414  ERAWBS  4397C  4411
4411  ERAWC   4408B
1782  ERB     1777B
1787  ERC     1781B
4612  ERCK    3511B  3580B  4110B
4620  ERCKA   4615B  4617B
4621  ERCKB   4619B
4664  ERCKBS  4612C  4647
4642  ERCKC   4621B
4645  ERCKD   4641B
4652  ERCKE   4646B
4662  ERCKF   4655B  4657B  4660B
4638  ERCKG   4624B  4626B  4628B  4630B  4632B  4634B
4640  ERCKH   4637B
4641  ERCKI   4639B
5281  ERCT    1268C  3012   3807C  3813
5282  ERDRIV  1481C  3808C  3866
5283  ERFILE  1484C  3810C  3867
5284  ERLNTH  1287
5285  ERREC   1487C  3812C  3868
5286  ERRFLG  3859
5287  ERTB    1287   1289   1467   1511   5268
5288  ERTBPT  1290C  3819   3849C
5394  EVNPAR  2456
5289  FAIL    4965C  4973C  4980
1864  FF      5547
3287  FFA     3289B
1898  FFB     1894B
3284  FFF     5456
1895  FFFA    1890B
2572  FI
2571  FILE    5552
2587  FILEA   2578B
2590  FILEB   2586B
2585  FILEC   2583B
5290  FILES   2965   2969   3643   3645   3649
5291  FIRM    1081C  1167   2169   4987
1078  FIRM10  1079B
5292  FLAG    1195C  1205   1274   1300   1303
1796  FM      1799B  5543
1802  FMA     1806B
1816  FMB     1811B
1819  FMC     1815B
3602  FNER    1108B  1123B  1137B  1428B  1701B  1718B  1735B  1755B  1782B  1816B
              1838B  1857B  1876B  1895B  1914B  1926B  1938B  1950B  1961B  1973B
              2046B  2051B  2073B  2128B  2139B  2155B  2180B  2201B  2222B  2273B
              2281B  2313B  2339B  2362B  2392B  2412B  2430B  2474B  2498B  2534B
              2556B  2587B  3030B  3539B  3546B  3661B  3958B  4007B  4092B  4266B
              4409B  4443B  4491B  4526B  4563B  4599B  4603B  4642B  4707B  4715B
              4727B  4735B  4751B  4846B  4955B  4974B  4999B  5015B  5050B  5071B
              5085B  5100B  5162B  5193B  5204B  5218B  5221B
4600  FNER01
4604  FNER02
3540  FNER03
1429  FNER04
1109  FNER05
1124  FNER06
4643  FNER07
3547  FNER08
4649  FNER09
1702  FNER10
1719  FNER11
1736  FNER12
1756  FNER13
2588  FNER14
4008  FNER15
4093  FNER16
3031  FNER17
3959  FNER18
1783  FNER19
5086  FNER20
```

```
1817  FNER21
5101  FNER22
5051  FNER24
1839  FNER25
1915  FNER26
2202  FNER27
2223  FNER28
1858  FNER29
1877  FNER30
1896  FNER31
1927  FNER32
1939  FNER33
1951  FNER34
1962  FNER35
2047  FNER36
2052  FNER37
2074  FNER38
4708  FNER39
4716  FNER40
4728  FNER41
4736  FNER42
4752  FNER43
1974  FNER44
2129  FNER45
2140  FNER46
2156  FNER47
2274  FNER48
2282  FNER49
2314  FNER50
2340  FNER51
2363  FNER52
2393  FNER53
2413  FNER54
2431  FNER55
2475  FNER56
2499  FNER57
2535  FNER58
2559  FNER59
5163  FNER60
5194  FNER61
5205  FNER62
5219  FNER63
5222  FNER64
2181  FNER65
5662  FNER66
5072  FNER67
3853  FNERA    3815B
3862  FNERB    3860B
3881  FNERB5   3804C   3858C   3878
3877  FNERC    3854B
3878  FNERD
4293  FNERDA
4267  FNERDD
3866  FNERE    3863B
4410  FNERE1
3864  FNERF    1478C   3806C   4292C
4444  FNERF1
4527  FNERF4
3872  FNERG    3870
3858  FNERMP   1500B
1138  FNERND
4975  FNERP0   4967C   4977C
5000  FNERQL   4998C
4492  FNERR1
4564  FNERR4
5016  FNERT0
4956  FNERT2
1846  FR       5545
1857  FRA      1852B
1860  FRB      1856B
5130  FRB33    2258B   2291B   2319B   4795B
5135  FRB3B5   5130C   5132
4468  FSF      1886B   4426B
4475  FSFB5    4468C   4472
4453  FSFEA    4455B
4464  FSFEB5   4452C   4461
4461  FSFEC    4458B
4459  FSFED    4460B
4452  FSFEOT   2895B   2949B
4425  FSFW     3593B   4379B
4443  FSFWA    4432B
4445  FSFWB    4442B
4448  FSFWB5   4425C   4445
4441  FSFWC    4439B
4440  FSFWD    4437B
4504  FSR      1848B   2403B   2421B   4480B
4510  FSRB5    4504C   4507
4479  FSRW     2921B   3557B   3913B   3923B   4453B
4491  FSRWA    4486B
4493  FSRWB    4490B
4500  FSRWB5   4479C   4497
4497  FSRWC    4494B
4016  FWB      3961B
4031  FWBA     4033B
4047  FWBB     4022B
4067  FWBB5    4018C   4064
4059  FWBC     4034B   4043B
4060  FWBD     4062B
4051  FWBE     4055B
4038  FWBF     4024B   4026B
4039  FWBG     4042B
5360  FWDFIL   4471
5378  FWDREC   4506
1147  GETBDC   1136B   1139B
4419  GETRNG   1546B   1909B   2264B   2297B   2325B   2465B   2490B   3600B   4420B   4844B
3294  GG       5457
3296  GGA      3300B
5117  GOBOT    1931B   2189B   2207B   2233B   2246B   2257B   2290B   2318B   2348B   2374B
                2398B   2418B   2455B   2479B   2541B   2574B
5120  GOBOTA   5121B
5293  GOFLAG    991    1034
4757  GOWRAP   1550B   1557B   1564B   1571B
3757  GRNA     3759B   3766B
```

```
3755  GRNADR  2776B  2778B  3323B  3339B
3763  GRNB    3765B
3772  GRNB5   3755C  3769
3768  GRNC    3761B
4674  HOOBF   1993
5294  IDEN
3305  II      5459
3307  IIA     3310B
5430  INBDC   1063   1076
4867  INCBA   4877B
4874  INCBB   4871B
4877  INCBC   4869B  4873B  4875B
4864  INCBFR  1601B  1625B  1636B  1652B  1663B
5431  INCONF
5399  INDEX   2510   3075   4816
5432  INFIRM  1064   1078
5433  INIDEN  1062   1068   5064
5036  INIZ    1313B  1543B  2090B  2561B  4845B  5038B
5039  INIZA   5040B
5048  INIZB
5056  INIZB5  5036C  5053
5053  INIZC   5049B
5434  INQLT2  4937
5435  INQLT3  4945
5436  INQLTJ  4990
5437  INRANG  4419   4806   5160   5176
5438  INRUPT  5039
3745  INSTAT  1693B  1707B  1711B  1723B  1830B  1849B  1868B  1887B  2120B  2131B
              2147B  2165B  2193B  2214B  2265B  2298B  2326B  2354B  2384B  2404B
              2422B  2466B  2491B  2531B  2549B  2575B  3253B  3401B  3509B  3518B
              3526B  3542B  3579B  3746B  3955B  3996B  4089B  4400B  4428B  4482B
              4517B  4554B  5122B  5163B  5203B
5439  INSTW1  1416   1744   1770   1804   1936   1959   3747   4597
5440  INSTW2  1066   1085   1726   1749   1773   1807   1942   1965   3745
5441  INTASK
5442  INWAIT  1065   1082   4829
5400  INZBDC  3252   4927   5037
5420  IOREAD  1043   2260   2293   2321   2350   2486   3505   4102
5421  IOWRIT  1905   2209   2380   2460   2545   3575   3988   4797
4688  ISAKDV  1991   2019   4702
4689  ISAKM1
4690  ISAKM2
4691  ISAKP   1996C  2061C  2081C
4692  ISAKS
4687  ISAKTL
4679  ISATDV  2014   2084
4750  ISATIM  4753B
4680  ISATM1
4681  ISATM2
4682  ISATP   4696C  4697C
4683  ISATS
4678  ISATTL
4227  ISB     3269B  4117B  4852B
4246  ISBA    4243B
4250  ISBB    4232B  4234B  4237B  4240B  4245B
4312  ISBB5   4227C  4271
4251  ISBC    4259B
4257  ISBD    4254B
4269  ISBE    4265B
4277  ISBF    4270B  4324B
4290  ISBG    4255B
4313  ISBGB5  4290C  4329
4303  ISBM    4301B
4311  ISBIS   4229C  4252
4310  ISBSB   3941C  4115   4228C  4251
3747  ISTATI  3746B
3316  JJ      5460
5295  JJPT    3213   3268   3285   3295   3306   3324C  3342C  3465   3467   3485
              3575
3332  KK      5461
5296  KKPT    3246   3267   3326C  3340C  3473   3475   3485   3493   3505
3667  LABEL   1016B  1040B  1341B  1553B  1560B  1567B  1575B  1595B  1619B  1646B
              1673B  1691B  1762B  1796B  1827B  1846B  1865B  1884B  1903B  1984B
              2108B  2187B  2231B  2288B  2346B  2368B  2435B  2503B  2539B  2571B
              2595B  2645B  2653B  2661B  2669B  2677B  2685B  2693B  2701B  2709B
              2717B  2725B  2733B  2741B  2750B  2783B  2813B  2832B  2860B  2877B
              2891B  2981B  3056B  3259B  3275B  3398B  3431B  3491B  3524B  3535B
              3555B  3564B  3591B  4904B
5297  LABEL1  1415C  1472C  3668C  3783   3821   3865   3962   3965   4235   4277
              4362   4613   4622   4653   4902   5020C  5184   5187
5298  LABEL2  1475C  3803C  3824   4294C
5401  LCHN0   2509   3074   4815
5402  LCHN1
5403  LCHN2
5404  LCHN3
4784  LDBUF   1578B  4764B  4774B
4786  LDBUFA  4788B
5299  LENGTH  3501   4099   5131
5199  LEVV10  2600   5210B
1011  LEVV15  1002
5215  LEVV5   5171   5233B
1006  LEVXA
1403  LEVXB
2007  LEVXC
2062  LEVXD
2079  LEVXE
4744  LEVXF
3678  LEVXG
5209  LEVXH
5232  LEVXI
2095  LEVXJ
5448  LINKS   3136   3169
5558  LIST    2926
3348  LL      5462
5451  LNKTAB  3137
4668  LVOC    2005
1368  MARW    1314B  1339B  1347B  1355B  1363B
1374  MARWA   1371B
1040  MI
3398  MM      5463
1254  MODEAD  1238B  1242B
3055  MODED   1240B  5448
3063  MODEDA  3064B
```

```
3094  MODEDB   3080B
3134  MODEDC   3145B   3153B   3155B   3157B
3138  MODEDD   3160B
3101  MODEDE   3090B
3150  MODEDF   3142B
3164  MODEDG   3144B   3146
3169  MODEDH   3165B
3170  MODEDI   3238B   3255B   3271B   3280B   3290B   3301B   3311B   3319B   3328B   3335B
               3344B   3386B   3404B   3427B   3435B   3478B   3487B   3514B   3520B   3531B
               3551B   3560B   3583B   3596B   3637B
3110  MODEDJ   3125B   3127B   5458
3117  MODEDK   3115B
3125  MODEDL   3122B
1440  MODEE    1222B
3642  MODEF    1224B
3647  MODEFA   3644B
3648  MODEFB   3646B
1038  MODEI    1226B
1045  MODEIA   1047B
1061  MODEIB   1129B
1068  MODEIC   1069B
1085  MODEID   1075B   1086B
1127  MODEIE   1073B   1095B   1112B   1141
1141  MODEIF   1057
1110  MODEIG   1107B
1116  MODEIH   1097B
1125  MODEII   1122B
1096  MODEIJ   1089B
1466  MODEP    1230B
1257  MODEKW   1244B   1246B
1445  MODEX    1234B
1450  MODEZ    1236B
1451  MODEZA   1462B
1453  MODEZB   1458B
1469  MPA      1513B
1511  MPB      1506B   1507
1514  MPDONE   1471B
5349  MSABRT   4302
5350  MSBYTE   4339
5351  MSCHAN   1038
5352  MSDENS   1087    1091    1093    1094    1102    1104    1105    1117    1119    1120
5353  MSDOWN   1090    1092
5354  MSEOT    3166    3647    5142
5355  MSERR    3013
5356  MSFILE   3021    3648
5357  MSFIRM   1154C   1161C   1166
5345  MSHEAD   3861
5358  MSINVL   1249
5359  MSIS     4348
5360  MSLINK   3134
5361  MSMODE   2998
5336  MSNAME    993
5362  MSNEXT   1193
5363  MSNRZI   1116    1118
5364  MSPARA   3110
5365  MSPARM   3439
5366  MSPASS   3004
5367  MSPE     1101    1103
5368  MSWLPS   1460
5369  MSRTC     997
5370  MSRTRY   4285    4661
5371  MSSB     4343
5372  MSSTAT   3601
      MTUS3       1    5575
5300  MXAD     2775C   3322C   3338C   3760
1260  MZIP     1246B
2981  NDPS     2954B   2962B   2970B   4246B
1183  NEXT     1201B   1250B   1332B   1441B   1446B   1509B   1514    3650B   3787    5113B
1188  NEXTA    1192B
1199  NEXTB    1198B
1196  NEXTC    1301B
1206  NEXTE    1202B
1300  NEXTF    1220B
1311  NEXTG    1304B   1312B
1318  NEXTH    1315
1214  NEXTI    1211B
1212  NEXTJ    1209B
1249  NEXTK    1213B
3408  NN       5464
3420  NNA      3418B
5388  NOPER    2146    2164
5395  NORMAL   1311    1685    2395    2480    3063
5405  NOTEST   2529    3103    4827
5301  NTST     3263C   3568C   3949C   4083C   4307B
5302  NULL     1287    4884    4907    5041    5554
5303  OFSETK   1586C   1604C   1628C   1655C   1679C   4766C   4803
5304  OFSETW   1585C   1603C   1627C   1653C   1678C   4765C   4796
4126  ONES     2647B   2655B   2663B   2671B   2679B   2687B   2695B   2703B   2711B   2719B
               2727B   2735B   2992B
4133  ONESA    4138
4134  ONESB    4127C
4170  ONESBS   4131C   4167
4135  ONESC    4129C
4154  ONESD    4145B   4149B
4155  ONESE    4159
4156  ONESF    4130C
4163  ONESG    4140    4161
4167  ONESM    4146B   4164B
3431  OO       5465
5423  OTCONF   1311    1685    2375    2395    2419    2456    2480    3063    4799
5424  OTCONT   1706    2035    2508    2509    2510    2515    2522    2523    2524    2525
               2527    2528    2529    3073    3074    3075    3084    3085    3086    3087
               3088    3094    3095    3096    3097    3101    3102    3103    3252    4808
               4815    4816    4818    4819    4820    4821    4822    4823    4825    4826
               4827    4927    5037    5216
5425  OTKOPT   1536    2028    2573    2607    2611    2984    3055    4368    4459    4541
5426  OTSTW1   2111    2142    2158
5427  OTSTW2   2113    2160
5428  OTTASK   1544    1699    1740    1764    1798    1907    1932    1955    2146    2164
               2191    2211    2262    2295    2323    2352    2382    2462    2488    2547
               3026    3507    3537    3577    3990    4104    4376    4398    4471    4506
               4543    4578    4801    5120
5112  PCM      1228B
```

```
5305  PFLAG    1469    1505    3788    3791
4334  PISB     4328B
4339  PISBA    4336B
4357  PISBB5   4334C   4354
4673  PLVL     2032C   2058C   4700
3439  PP       5466
3466  PPA      3464
3474  PPB      3472
1166  PRMSG    1156B
5306  PTRN     3118    3129C   3130    3286    3425C   3458    3936C   4020
4905  QL
4927  QLTA     4928B
4925  QLTAA    5009B
5032  QLTB5    4901C   5021
4930  QLTC     4926B
4937  QLTD     4938B
4945  QLTE     4932B   4946B
4952  QLTF     4941B
5307  QLTFLG   4911C   4925    4930    5013
4961  QLTG     4954B
4971  QLTH     4979B
5308  QLTI     4940C   4948C   4961
4976  QLTJ     4972B
4990  QLTK     4991B
5004  QLTL     4936B   4957B   4981B   4986B   4989B   4994B   5026
5017  QLTM     5014B
4901  QLTRUN   1168B   1453B
5026  QLTTH    4920
3482  QQ       5467
5309  R3HOLD   1302    3777C
5310  RANADR   3756    3768C
5311  RANDOM   2898    2904
5312  RANGE    2279    2299    2327    3624    4419    4846    5160    5178
5252  RBUF     2251    2260    2276    2293    2303    2321    2329    2350    2484    2486
               2637    2766    2788    2865    3325    3501    4097    4157    4214    4804
               4851    5131    5296
5382  READFD   2262    2295    2323    2488    3507    4104
5386  READRV   2191
1172  RESTRT   992B    3587B   3795
1034  RESUME   3676
5361  REVFIL   1955    4543
5379  REVREC   1932    4578
5376  REWIND   1699    1740    3028    3537    5120
4174  RLD      2743B   2752B
4175  RLDA     4195
4185  RLDB     4183B
4223  RLDB5    4174C   4220
4192  RLDC     4185C
4211  RLDD     4202B   4206B
4212  RLDE     4216
4213  RLDF     4186C
4220  RLDG     4197    4203B   4218
5313  RNG      1549C   1556C   1563C   1570C   1599C   1606C   1611C   1623C   1630C   1634C
               1637C   1638C   1649C   1657C   1661C   1676C   2459C   2483C   3116C   3117
               3128C   3231    3288    3299    3309    3420C   3452    3482    3505    3575
               3936C   3988    4032    4041    4054    4072C   4102    4258    4833    4865
5314  RNGFLG   1194C   3114    3124C   4616
1982  RP       5549
1992  RPB      1994B
2102  RPB6     1982C   2098
2001  RPC      2087B
2017  RPD      2013B
2021  RPE      2016B
2025  RPF      2086B
2028  RPFA     2029B
2048  RPJ      2042B
2053  RPK      2049B
2063  RPK1     2060    4743B
2075  RPK2     2071B
2080  RPK3     2076B
2089  RPM
2086  RPN      2026B   2040B   2044B   2053B   2055B
2098  RPU      2091
2101  RPK3     1983C   2045    2050    2072    2089    4706    4714    4726    4734    4750
3807  RPRT     4319B
3491  RR       5468
3501  RRA      3495C
3505  RRB      3506B
3507  RRC      3508B
3512  RRD      3496
4071  RREC     2249B   2635B   2817B   2863B   4155B   4212B
4095  RRECB    4091B
4122  RRECB5   4085C   4119
4101  RRECBA   4096B
4097  RRECBB   4075C
4102  RRECC    4103B
4104  RRECD    4105B
3685  RSET     4279B   4281B   4284B   4286B
3906  RT2      5539    5551
3908  RT4      2241    2252    3264    3498    3569    4282    4303    4658    4759
3910  RT5      2618    2795    2843    4137    4194
3918  RT5B5    3910C   3915
3920  RT6      2638    2820    2866    4156    4215
3927  RT6B5    3920C   3924
1028  RTCFA
3673  RTCFC    1011
5315  RTCHZ    998     1020
5316  RTCN     3897C   4297C   4298
3685  RTCNB
5317  RTFL     3886C   3899C   4295    4645
5318  RTRN     3265C   3499C   3570C   3889B   3943C   4077C   4283    4304C   4652B   4659
               4760C
5319  RTRY     1590C   1614C   1641C   1668C   1683C   3262C   3497C   3567C   3890B   3945C
               4079C   4662B   4762C   4772C
1691  RW       5541
1704  RWA      1700B
1707  RWAA     1709B
3729  RWAQ     2625B   2802B   2850B   2879B   2935B   2955B   2987B   3036B   3977B   4144B
               4201B   4241B   4370B
3741  RWAQZ    3733B   3736B   3739B
1718  RWB      1713B
1723  RWC      1717B   1725B
1726  RWCC     1727B
```

```
1735  RWD      1730B
1740  RWE      1695B   1734B   1741B
1744  RWF      1745B
1749  RWFF     1750B
1755  RWG      1746B
1758  RWH      1754B
2108  SA       5550
5490  SA10DV   2598
5491  SA10M1
5492  SA10M2
5493  SA10P    2601C
5494  SA10S
5489  SA10TL
5499  SA15DV   1004    1396    2093
5500  SA15M1
5501  SA15M2
5502  SA15P    1003C   1389C   2092C   3677C   5208C   5231C
5503  SA15S
5498  SA15TL
5481  SA5DV    1013    5169
5482  SA5M1
5483  SA5M2
5484  SA5P     1012C   5172C
5485  SA5S
5480  SA5TL
2111  SAA      2112B
2113  SAB      2114B
2126  SAC      2123B
2131  SAD      2127B
2139  SAE      2134B
2142  SAF      2136B   2143B
2156  SAG      2154B   2159B
2155  SAH      2150B
2160  SAI      2161B
2180  SAJ      2175B
2174  SAK      2168B   2171B
2175  SAL      2173B
5320  SAVE1    4864C   4878
2187  SB       2179B
2201  SBA      2196B
2207  SBB      2200B   2210B
2211  SBC      2212B
2222  SBD      2217B
2231  SC       2221B
2260  SCA      2261B
2262  SCB      2263B
2273  SCC      2268B
2276  SCD      2272B
2261  SCE      2278B
2257  SCF      2253
5406  SCRDY    2527    3101    4825
2288  SD       2280B
2293  SDA      2294B
2295  SDB      2296B
2313  SDC      2301B   2305B   2308B
2318  SDD      2312B
2321  SDE      2322B
2323  SDF      2324B
2339  SDG      2328B   2331B   2334B
2346  SE       2242    2244    2255    2338B   2351B
2352  SEA      2353B
2362  SEB      2357B
3795  SENSA    3785B
3796  SENSB    3781B   3789B   3793B
3798  SENSB5   1177C   1305    1508C   1515C   3776C   3779    3792
3776  SENSE    1325B   1454B   1790B   1822B   1841B   1860B   1879B   1898B   1976B   2562B
               3234B   3237B   3254B   3270B   3278B   3327B   3343B   3385B   3402B   3426B
               3434B   3477B   3486B   3512B   3519B   3530B   3550B   3558B   3581B   3594B
               3636B   3677B   4010B   4111B   4353B
5551  SEQ1     1257    1373
5539  SEQN     1254
3894  SET      1588B   1612B   1639B   1666B   1681B   2248B   2633B   2816B   2862B   3266B
               3503B   4154B   4211B   4305B   4758B   4770B
4390  SETCHA   4392B
4387  SETCHN   1412B   3109B   4913B   5007B
2366  SF       2361B   2376B
2380  SFA      2381B
2382  SFB      2383B
2392  SFC      2387B
2395  SFD      2391B   2396B
2412  SFE      2407B
2418  SFF      2411B   2420B
2430  SFG      2425B
2374  SFH      2371B
2435  SG       2429B
2442  SGA      2444B
2446  SGB      2448B
2456  SGC      2457B   2461B
2462  SGD      2463B
2474  SGE      2469B
2479  SGF      2473B   2481B
2486  SGG      2487B
2488  SGH      2489B
2498  SGI      2494B
2503  SH       2373B   2497B
2539  SI       2506B   2533B
2545  SIA      2546B
2547  SIB      2548B
2558  SIC      2552B
2561  SID      2557B
5321  SLDB     1490C   2440C   2485C   2544C   3494C   3837    3871    3873    3940C   3971
               3988    4019    4074C   4102    4116    4908C   5042C
3518  SS       5469
 986  START     989    1030B   1232B   3659    5575
1002  STARTA    996B
5322  STAT1    1416    1418    1420    1493C   1694    1712    1724    1728    1744    1746
               1770    1775    1804    1809    1831    1850    1869    1888    1912    1922
               1936    1943    1959    1966    2116C   2121    2132    2148    2166    2194
               2215    2266    2306    2332    2355    2385    2405    2423    2467    2492
               2550    2576    3527    3543    3629    3747    3840    3874    3956    3997
               4000    4090    4401    4429    4454    4483    4493    4518    4555    4565
               4597    4601    4618    4620    4909C   5043C   5140
5323  STAT2    1085    1088    1096    1496C   1708    1714    1726    1731    1749    1751
```

		1773	1778	1807	1812	1634	1853	1872	1891	1942	1946
		1965	1969	2117C	2124	2135	2151	2176	2197	2218	2269
		2309	2335	2358	2388	2408	2426	2470	2495	2532	2553
		2579	3634	3745	3843	3875	4003	4405	4433	4487	4522
		4559	4638	4640	4910C	5044C					
5407	STOPIO	1706	2035	5216							
5324	SUPRES	1440C	1445	3853	4280	4300	4335	4656			
3684	SYNCH	1027B									
3686	SYNCHA	3687B									
3690	SYNCHB	3692B	3697B	3707B							
3700	SYNCHC	3675C	3696	3706							
1536	T1	5540									
1541	T1A	1538B									
1544	T1B	1545B									
1553	T2										
1560	T3										
1567	T4										
1575	T5										
1581	T5A	1584B									
1595	T6	1589									
1619	T7	1613									
1646	T8	1640									
1673	T9	1667									
1685	T9A	1539B	1682	1686B							
5325	TEMP	5326	5327	5328							
5326	TEMPA	1056C	1074	1135	1196	1207	1216	1293	1503C	1504	3006C
		3007	3019C	3020	3119	3120	3121	3138	3139C	3140	3150C
		3151	3218C	3219	3226C	3227	3296	3297	3409C	3410	3445C
		3446	3484C	3485	3605C	3606	3612	3614	4051	4052	4340C
		4341	4342	4344	4963C	4971	4995C	4996	5039	5048	
5327	TEMPB	1070C	1106	1121	3217C	3219	3410	3414	3424	3619	4349
		4903C	5019								
5328	TEMPC										
5408	TESTMD	2508	3073	4807							
3659	TH15	1133	5017								
1018	TI										
5064	TIMCA	5065B									
5076	TIMCB	5068B									
5108	TIMCB5	5062C	5105								
5085	TIMCC	5080B									
5091	TIMCD	5070B									
5100	TIMCE	5095B									
5105	TIMCF	5084B	5087B	5099B							
5061	TIMCHK	1787B	1819B								
5329	TIMCNT	1767C	1771C	1801C	1805C	5079	5083	5094	5098		
3704	TIME	1704B	1768B	1802B	1934B	1957B	2118B	2144B	2162B	2517B	5181B
5534	TIMER	1786									
5535	TIMFM	1820									
5507	TSA1	1053	1055C	1174	4917	4918C					
5514	TSA1B3										
5512	TSA1EA										
5510	TSA1F										
5508	TSA1I										
5513	TSA1P	1142C	5027C								
5509	TSA1R3										
5511	TSA1Z										
5518	TSA2										
5525	TSA2B3										
5523	TSA2EA										
5521	TSA2F										
5519	TSA2I										
5524	TSA2P										
5520	TSA2R3										
5522	TSA2Z										
3524	TT	5470									
3530	TTA	3528B									
4794	TURN	1567B	1607B	1631B	1658B	1680B	4767B	4775B			
4797	TURNA	4798B									
4801	TURNB	4802B									
4839	TURNB5	4794C	4836								
4809	TURNC	4810B									
4814	TURND	4834B									
4829	TURNE	4830B									
4799	TURNF	4800B									
5377	UNLOAD										
3535	UU	5471									
3542	UUA	3538B									
3548	UUB	3545B									
3550	UUC	3541B									
3711	VDTR	1169B	1191B	3000B	3222B	4346B	4351B	4588B			
3717	VDTRA	3712C	3713								
3719	VDTRB5	3711C	3714								
5330	VRBL	3227	3228	3446	3447	3452	3453	3458	3459	4344	4345
		4349	4350	4996	4997						
3555	VV	5472									
5158	WAIT	3995B	4109B	4377B	4399B	4427B	4481B	4516B	4553B		
5165	WAITA	5161B									
5178	WAITB	5179B									
5242	WAITB5	5158C	5239								
5221	WAITC	5217B									
5239	WAITD	5186B	5189B	5192B	5207						
5224	WAITE	5195B									
5235	WAITF	5229B	5230								
5207	WAITG	5202B									
5409	WAITLP	1082	2528	3102	4826						
5251	WBUF	1581C	1597	1609C	1621	1632	1651	1659	1665C	1677	1905
		2209	2240	2379C	2380	2439	2460	2617	2768	2794	2836
		3341	4136	4178	4179	4193	4785	4797	4850	5295	
4361	WFM	2397B	2590B	2883B	2886B	3400B					
4379	WFMB	4371B									
4383	WFMB5	4361C	4380								
4380	WFMC	4378B									
4368	WFMD	4369B									
4370	WFME	4364B	4367B								
4770	WRAPA	4761									
4777	WRAPB	4771									
4780	WRAPB5	4757C	4777								
3935	WREC	2237B	2614B	2791B	2839B	4133B	4190B				
3961	WRECA	3957B									
3977	WRECB	3964B	3967B								
4014	WRECB5	3951C	4011								
3983	WRECC	3978B									
3987	WRECD	3979B	3982B								

```
 3988  wRECE     3989B
 3990  wRECF     3991B
 4000  wRECG     3998B
 4007  wRECH     4002B
 4010  wRECI     3983B   4006B
 5410  wRENB     2515
 5413  wRINM     2524
 5385  wRITE     1907    2211    2352    2382    2462    2547    3577    3990
 5384  wRITFM    1798    4376
 5387  wRNGGU    4801
 1082  wTLP      1083B
 3564  WW        5473
 3575  wWA       3576B
 3577  wWB       3578B
 3581  wWC       3566
 1325  X0        1176    1321    1359B   1364B
 1341  X1        1318    1335B
 1349  X2        1319    1343B
 1357  X3        1320    1351B
 5331  X3333     2277    3501    4098    5131
 5411  XFER1     2522
 5412  XFER2     2523
 1328  XOA       1331B
 1333  XOB       1330B
 3587  XX        5474
 3591  YY        5475
 5341  Z1        5337
 5343  Z2        5339
  984  ZERO       988
       ZHCOMM     985    2760    3317    3333
       ZHIAFB     986    2001C   2002C   2003C   2004C   2010    2067    4719    4732
       ZHISAZ     985    1005C   1014C   1386    1387    1391    1392    1397C   1408C   1409C
                 1990    2015C   2020C   2085C   2094C   2599C   4674    5170C   5224    5225C
                 5235C
       ZHNTSA     985    1054C   1175C   4919C
       ZHRTCC     985    1024C   3686    5166C
       ZHRTCI     985    1023C   5167C
       ZHRTCL     985    1026C   1185C
       ZHTH15     985    1058C   1134C   4921C   5018C
       ZV$BKF     985    3780
       ZV$BRK    3778B
       ZV$ER      986    3862B
       ZV$F       986    1287B   3501B   4095B   5131B
       ZV$FR     1677B   2898B   3296B   3410B   4051B   4178B
       ZV$FRB     986    4049C   4177C
       ZV$HA     3227B   3446B   3452B   3458B   4344B   4348B   4349B   4996B
       ZV$HM     2542B
       ZV$HR      985    2759    3316    3332
       ZV$HRL     986
       ZV$HRU     986
       ZV$HZ     3458B   4344B   4349B
       ZV$IA     1196B   3138B
       ZV$ID      998B
       ZV$IH      998B   1039B   3120B   3649B
       ZV$LR      986    3758    3764
       ZV$MD     3465B
       ZV$PCH    5112B
       ZV$Q      1193B   3648B
       ZV$QC      997B   1038B   3110B   3134B
       ZV$RD      993B   2542B
       ZV$SV1     986
       ZV$SV2     986
       ZV$T       997B   1038B   1094B   1105B   1120B   1166B   1193B   1249B   1460B   2998B
                 3004B   3013B   3021B   3110B   3134B   3166B   3230B   3439B   3447B   3453B
                 3459B   3601B   3647B   3648B   3713B   3861B   4285B   4302B   4339B   4343B
                 4348B   4661B   4884B   5142B
       ZV$TC     1094B   1105B   1120B   1166B   1249B   1460B   2998B   3166B   3439B   3601B
                 3861B   4339B   4884B   5142B
       ZV$TD     1461B   1504B   3007B   3012B   3606B   3866B
       ZV$TH     1167B   1461B   1504B   3007B   3012B   3020B   3219B   3465B   3467B   3473B
                 3475B   3606B   3614B   3619B   3624B   3629B   3634B   3645B   3866B   3867B
                 3868B   3871B   3873B   3874B   3875B   4341B
       ZV$THZ    3467B   3475B   3629B   3634B   3873B   3874B   3875B
  5332 ZVSTAT    1196    1197    3138
  3600 ZZ        5476
 981 LABELS
4674 REFERENCES
5575 RECORDS
   0 U FLAGS
   0 M FLAGS
 170 N FLAGS
6 CROSS REF VERSION L - 24 SEPT, 1976
RS LINKER VERSION 5.00 06/29/78  1830.4 EDT THU
LINK MAP FOR MTUS3
 START          0100
 LOW            0000
 HIGH           2C52
 CURRENT        2C53
*LOC DEFS
  ZHCOMM        0000
  *MTUS3        0000   REV G
  ZHPFR         0000
  ZHTSA         0002
  ZHNTSA        0010
  ZHRTCI        0014
  ZHRTCC        0015
  ZHRTCL        0016
  ZHWDTC        0017
  ZHMERC        001F
  ZHIAFB        0020
  ZHTH29        0063
  ZHTH28        0064
  ZHTH27        0065
  ZHTH26        0066
  ZHTH25        0067
  ZHTH24        0068
  ZHTH23        0069
  ZHTH22        006A
  ZHTH21        006B
  ZHTH20        006C
  ZHTH19        006D
  ZHTH18        006E
  ZHTH17        006F
```

```
ZHMEMP      006F
ZHTH16      0070
ZHLERR      0070
ZHTH15      0071
ZHNRES      0071
ZHTH14      0072
ZHPMEM      0072
ZHTH13      0073
ZHP-OP      0073
ZHTH12      0074
ZHTH11      0075
ZHTH10      0076
ZHTH9       0077
ZHTH8       0078
ZHTH7       0079
ZHTH6       007A
ZHOVFL      007A
ZHTH5       007B
ZHOP-N      007B
ZHTH4       007C
ZHTH3       007D
ZHSC-N      007D
ZHTH2       007E
ZHTRC       007E
ZHTH1       007F
ZHMCL       007F
ZHISA2      0080
ZHIVDS      0080
ZHIVBS      0080
*ZVST       2343  REV. 5.0
 ZVSGC      2360
 ZVSTC      234C
 ZVSG       2355
 ZVST       2343
*ZVSIH      2374
 ZVSID      2379
 ZVSIH      2374
 ZVSIAD     237E
 ZVS---2    2396
 ZVS---3    23A8
*ZVSTH      240D
 ZVSTH      240D
 ZVSTD      2442
 ZVSTH2     2435
*ZVSIA      245D  REV. 7
 ZVSIA      2460
 ZVSARG     250F
 ZVSADF     2511
 ZVS---1    24CC
 ZVSIAV     245E
*ZVSF       251C
 ZVSF       251C
*ZVSFR      252A
 ZVSFR      252A
 ZVSFRD     257E
 ZVSFI      254C
 ZVSFS      256F
 ZVSFRA     257C
 ZVSFRX     257D
 ZVSFRR     2541
 ZVSFRM     257B
*ZVSHA      2581
 ZVSHA      2581
 ZVSHZ      2586
 ZVSHS      2586
*ZVSMB      25BA
 ZVSMB      25BA
*ZVSBRR     25DA
 ZVSBRR     25DA
*ZVSPCH     25F4
 ZVSPCH     25F4
*ZVSGP      26F6
 ZVSGP      26F6
 ZVS---4    2716
*ZVSHD      2722
 ZVSHD      2722
*ZVSER      2754  REV. 5.0
 ZVSER      2754
 ZVSTA      2780
 ZVS---0    2767
*ZVSRD      27C4  REV. 7
 ZVSRD      27C4
 ZVSHM      283A
 ZVSHR      27F3
 ZVSLR      27F0
 ZVSBKF     27EC
 ZVSSV2     29A9
 ZVSGTP     26DB
 ZVSSV1     2999
 ZVSSV3     29B9
 ZVSAF      27D5
 ZVSTTY     27D7
 ZVSTID     27D6
 ZVSCF2     27E0
 ZVSTK      27DC
 ZVSRAR     27DD
 ZVSSTI     27E1
 ZVSRCC     27E2
 ZVSBUD     27D8
 ZVSOLB     27E4
 ZVSRCB     27E5
 ZVSNSR     27E9
 ZVSSTR     27E7
 ZVSBRS     27EB
 ZVSIZ      27FE
 ZVSDAT     27D3
 ZVSHRU     27ED
 ZVSHRL     27EE
 ZVSLRU     27EF
 ZVSLRL     27F0
 ZVSHBD     27F1
 ZVSCF1     27DF
 ZVS---5    27F6
```

```
ZVSRMD      27D4
ZVSMCP      27F2
HIBAUD      27F1
ZVSRAW      27DE
ZVSRDT      29F5
ZVSCTL      27DB
ZVSBI       2916
ZVSTST      2A4B
ZVSMDC      2A1F
ZVSR99      2C1D
ZVS1SA      27F9
ZVSUIN      27F4
ZVS2RD      2878
ZVSBSM      287A
ZVSCPD      27DA
ZVSR50      2858
ZVSR60      2863
ZVSRT       2B5A
ZVSALL      27D9
*MLCHPG     2C22   T+V
 MLCHPG     2C22
 ENDCHP     2C53
```