```
000001                          TITLE  CPFS4, 'REV A'
000002                  *   WRITABLE CONTROL STORE (WCS) TEST
000003                  *   PART NO.
000004                  * CPFX4  60135116-001
000005                  * CPFS4  60135117-001
000006                  * CPFL4  60135118-001
000007                  *
000008                  * DESCRIPTION
000009                  * -----------
000010                  *   THIS T&V PROGRAM VERIFIES PROPER OPERATION OF THE LEVEL 6 WRITABLE
000011                  *   CONTROL STORE (WCS).  IT PROVIDES A FIRST LEVEL OF DIAGNOSIS WHEN
000012                  *   FAILURES ARE DETECTED, AND MAKES FACILITIES AVAILABLE TO SUPPORT
000013                  *   EXTENSIVE PROBLEM INVESTIGATIONS.
000014                  *
000015                  *
000016                  * REVISION HISTORY
000017                  * -------- -------
000018                  *  A    MAR 1978   ORIGINAL RELEASE
000019                  *
000020                  *
000021                  *     THIS DOCUMENT AND THE INFORMATION CONTAINED THEREIN IS CONFIDENTIAL AND
000022                  *  PROPRIETARY TO AND THE EXCLUSIVE PROPERTY OF HONEYWELL INFORMATION SYSTEMS
000023                  *  INC.   IT  IS MADE AVAILABLE ONLY  TO  HONEYWELL AUTHORIZED RECIPIENTS FOR
000024                  *  THEIR USE SOLELY  IN  THE MAINTENANCE AND OPERATION OF HONEYWELL PRODUCTS.
000025                  *  THIS DOCUMENT AND INFORMATION MUST BE MAINTAINED  IN STRICTEST CONFIDENCE;
000026                  *  IT MUST NOT BE REPRODUCED IN  WHOLE OR IN PART;   AND IT SHALL NOT BE DIS-
000027                  *  CLOSED TO ANY OTHER PARTY WITHOUT THE PRIOR WRITTEN CONSENT OF HONEYWELL.
000028                  ***************************************************************************
```

```
000029                          / PROGRAM PREPARATION
000030                          * ------- -----------
000031                          *   THE ROOT SOURCE OF THIS PROGRAM, AFTER THE ADDITION OF THE APPROPRIATE
000032                          *   TITLE AND END STATEMENTS, WAS PROCESSED BY THE HOST RESIDENT ASSEMBLER
000033                          *   TO CREATE EITHER SHORT OR LONG ADDRESS FORM (SAF OR LAF) OBJECT TEXT
000034                          *   AND LISTING.  THE OBJECT TEXT WAS FURTHER PROCESSED BY THE HOST
000035                          *   RESIDENT LINKER USING THE APPROPRIATE CONSOLE ZV$LIB (VERIFICATION
000036                          *   SERVICE LIBRARY) TO CREATE A PUNCH SEGMENT CONTAINING AN EXECUTABLE
000037                          *   MODULE.  THE ASSEMBLY LISTING WAS AUGMENTED WITH CROSS-REFERENCE DATA
000038                          *   (XREF), PLUS THE LOAD MAP FROM THE LINKER TO CREATE A LIST SEGMENT.
000039                          *              ROOT          SAF           LAF
000040                          *              ----          ---           ---
000041                          *   NAME       CPFX4         CPFS4         CPFL4
000042                          *   DOCUMENT   60135116-001  60135117-001  60135118-001
000043                          *
000044                          * DISTRIBUTION
000045                          * ------------
000046                          *   THE ELEMENTARY ITEMS SUBMITTED TO FED SOFTWARE DISTRIBUTION, BY
000047                          *   ENGINEERING DESIGN PROCESSING, WERE THE EXECUTABLE LINKED IMAGES, ON
000048                          *   DISKETTE, OF GISS2 AND GISL2, AND MAGNETIC TAPE IMAGES OF THE
000049                          *   AUGMENTED LISTINGS.
000050                          *
000051                          *   REPRODUCTIONS OF THE EXECUTABLE LINKED IMAGES MAY BE AS DUPLICATE CARD
000052                          *   DECKS OR AS A MEMBER OF A MULTIPLE MEMBER FILE.  IN THE MOST FREQUENT
000053                          *   CASE, IT WILL BE FOUND AS MEMBER "SY" (SAF) OR "LY" (LAF) WITHIN FILE
000054                          *   "PROGFILE" OF DISKETTE VOLUME ENTITLED "DIAGS".
000055                          *
000056                          *   DISTRIBUTION OF  THE LISTINGS, WHICH SHOULD BE AVAILABLE IF ANY COMPLEX
000057                          *   MAINTENANCE OR REPAIR IS TO BE PERFORMED, IS NORMALLY AS A PRINTED
000058                          *   COPY.
000059                          *
000060                          * ROUTINE DEMONSTRATION
000061                          * ------- -------------
000062                          *   A MINIMUM SATISFACTORY TEST FOR NORMAL OPERATION MAY BE OBTAINED BY
000063                          *   RUNNING TWO PASSES.
000064                          *
000065                          * STORAGE
000066                          * -------
000067                          *   THIS PROGRAM REQUIRES AT LEAST 8K WORDS OF MAIN MEMORY.
000068                          *
000069                          * OPERATION
000070                          * ---------
000071                          *   LOAD AND START (OR RESTART) THE PROGRAM AT LOCATION HEX 0100. THE PROGRAM
000072                          *   IDENTIFICATION WILL BE DISPLAYED ON THE CONSOLE.  THE INITIAL START WILL
000073                          *   ALSO DISPLAY:
000074                          *
000075                          *     THE ZV$LIB REVISION NUMBER
000076                          *     THE ADDRESS FORM (SAF OR LAF)
000077                          *     I/O  EQUIPMENT DETECTED IN THE SYSTEM
000078                          *     MEMORY SIZE
000079                          *
000080                          *   THE ABOVE DISPLAY MUST BE VERIFIED BY THE OPERATOR.  IT IS OMITTED
000081                          *   ON RESTARTS.
000082                          *
000083                          *   ALSO DISPLAYED INITIALLY, AND DURING EACH RESTART AT HEX 0100:
000084                          *
000085                          *
000086                          *   THE FOLLOWING IS A TYPICAL RESULT OF LOADING AND RUNNING THE PROGRAM:
000087                          *
000088                          *     CPFS4   WCS TEST   -- MAR 1978   REV A
000089                          *     ZV$LIB  REV. 6.00
000090                          *
000091                          *     WDT
000092                          *     CHAN  DEVC  ID
000093                          *     0400  DSK1  2010
000094                          *     0480  DSK1  2010
000095                          *     0500  CONS  2018
000096                          *     0580  CDR   2008
000097                          *     2000  ISL   2402
000098                          *     MEMORY LOW  0000----
000099                          *     MEMORY HIGH 00007FFF  32K
000100                          *
000101                          *
000102                          *THE CONSOLE WILL NOW PRINT THE MESSAGE:
000103                          *
000104                          *
000105                          *    THE 1ST ERROR MESSAGE SHOULD ALWAYS BE CONSIDERED 1ST.  SUBSEQUENT
000106                          *    ERROR MESSAGES COULD BE CAUSED BY PREVIOUS ERRORS.
000107                          *
000108                          *
000109                          *   IF IN TRANSPARENT MODE, THE CONSOLE WILL PRINT:
000110                          *
000111                          *
000112                          *    TRANSPARENT MODE - HEX SWITCH SHOULD BE "0"
000113                          *
000114                          *
000115                          *   IF IN SEQUENTIAL MODE, THE CONSOLE WILL PRINT:
000116                          *
000117                          *
000118                          *    SEQUENTIAL MODE - HEX SWITCH SHOULD BE "F"
000119                          *
000120                          *
000121                          *   THEN,DEPENDING ON THE CONFIGURATION, THE CONSOLE WILL PRINT:
000122                          *
000123                          *
000124                          *    RAM NO. 1 PRESENT
000125                          *
000126                          *    OR
000127                          *
000128                          *    RAM NO. 2 PRESENT
000129                          *
000130                          *    OR
000131                          *
000132                          *    RAM NO. 1 PRESENT
000133                          *    RAM NO. 2 PRESENT
000134                          *
000135                          *
000136                          *   THE ABOVE MESSAGES MUST BE VERIFIED BY THE OPERATOR.
000137                          *
000138                          *
000139                          *   THE PROGRAM NOW IS TESTING THE RAM ARRAY BOARDS.  AFTER THE ARRAY BOARD
000140                          *   IS COMPLETELY TESTED, THE PROGRAM WILL START TESTING THE MOTHER BOARD.
000141                          *   DEPENDING ON THE MODE, A MESSAGE WILL BE PRINTED AS FOLLOWS:
```

```
000142                          *
000143                          *
000144                          *    TRANSPARENT MODE TEST NEXT, IF "O.K." NOT PRINTED IN 5 SEC, THEN MB
000145                          *    FAULT.  THIS MESSAGE WILL NOT BE PRINTED IN SUBSEQUENT PASSES!
000146                          *
000147                          *
000148                          *    OR
000149                          *
000150                          *    SEQUENTIAL MODE TEST NEXT, IF "O.K." NOT PRINTED IN 5 SC, THEN MB
000151                          *    FAULT.  THIS MESSAGE WILL NOT BE PRINTED IN SUBSEQUENT PASSES!
000152                          *
000153                          *
000154                          *
000155                          *
000156                          *
000157                          *
000158                          *
000159                          *
000160                          *    CONSOLE SEARCH RULES
000161                          *    ------- ------ -----
000162                          *     THE CONSOLE SEARCH RULES ARE:  FIND THE CONSOLE WITH THE LOWEST  CHANNEL
000163                          *     NUMBER CONNECTED THRU AN  MDC  CONTROLLER.  IF THERE IS NO CONSOLE ON AN
000164                          *     MDC, THEN SEARCH FOR A TERMINAL WITH THE HIGHEST CHANNEL NUMBER ASSIGNED
000165                          *     TO AN ACLA ADAPTER ON AN MLC CONTROLLER.  IF NO ASYNC ADAPTER  IS FOUND,
000166                          *     THEN GO TO THE FULL CONTROL PANEL.
000167                          *
000168                          *     THERE ARE THREE CONSOLE CHANNEL OPTIONS DETERMINED BY THE VALUE  OF  LO-
000169                          *     CATION "ZV$TTY".
000170                          *
000171                          *     IF ZV$TTY EQUALS (0000), SEARCH FOR A CONSOLE.
000172                          *     IF ZV$TTY EQUALS (FFFF), ASSUME THERE IS NO CONSOLE.
000173                          *     IF ZV$TTY EQUALS NEITHER (0000), NOR (FFFF), THEN IT IS THE CONSOLE CHAN-
000174                          *     NEL NUMBER.   NOTE:  DEFAULT IS TO SEARCH FOR A CONSOLE.
000175                          *
000176                          *     ALL CONSOLE I/O IS EVEN PARITY.  IF CONSOLE IS ON MLC,  IT MUST BE ASYNC
000177                          *     AND THE BAUD RATE SET AT 1200 TO MAICH THE PROGRAM SUPPLIED RATE.  IF IT
000178                          *     IS NECESSARY TO CHANGE THE PROGRAM BAUD RATE,   THEN THE  NEW  BAUD RATE
000179                          *     CODE SHOULD BE PUT INTO LOCATION "ZV$BUD" IN HEX.  THE TERMINAL BAUD RATE
000180                          *     MUST BE  SET  TO MATCH THIS NEW BAUD RATE.  THE CORRECT HEX VALUE MAY BE
000181                          *     OBTAINED FROM THE FOLLOWING TABLE.
000182                          *
000183                          *-----------------------------------------*
000184                          *                                         *
000185                          *        BAUD  RATE  TABLE                *
000186                          *                                         *
000187                          *-----------------------------------------*
000188                          * ACLA I.D.          (2118)(2110)        (2108)
000189                          * BAUD-RATE
000190                          *    50                 0                  1
000191                          *    75                 1                  2
000192                          *   110                 2                  3
000193                          *   134                 3                  4
000194                          *   150                 4                  5
000195                          *   200                 5                 ---
000196                          *   300                 6                  6
000197                          *   600                 7                  7
000198                          *   900                ---                 8
000199                          *  1050                 8                 ---
000200                          *  1200                 9                  9
000201                          *  1800                10 (A)              10 (A)
000202                          *  2000                11 (B)             ---
000203                          *  2400                12 (C)              11 (B)
000204                          *  3600                ---                 12 (C)
000205                          *  4800                13 (D)              13 (D)
000206                          *  7200                ---                 14 (E)
000207                          *  9600                14 (E)              15 (F)
000208                          * 19200                15 (F)             ---
000209                          *
000210                          *     TO MAKE ANY OF THE ABOVE CHANGES,  LOAD AND HALT THE PROGRAM BEFORE EX-
000211                          *     ECUTION.  INSERT CHANGE THEN EXECUTE.  MEMORY LOCATIONS OF "ZV$TTY" AND
000212                          *     "ZV$BUD" MAY BE FOUND IN MAP AT END OF LISTING.
000213                          *     CONSULT LEVEL-6 T&V MANUAL "AW94" FOR DETAILS ON HOW TO LOAD THE TESTS.
000214                          *
000215                          * END-OF-PASS REPORT
000216                          * ----------------
000217                          *   EACH TIME ALL WCS-RELATED LOGIC HAS BEEN TESTED, AN END-OF-PASS
000218                          *   MESSAGE WILL BE REPORTED ON THE CONSOLE (IF PRESENT).  THE PASS
000219                          *   THE PASS COUNT MAY ALSO BE OBSERVED IN B4 (IN DECIMAL).
000220                          *
000221                          *   EVERY 3 MINUTES, THE PROGRAM WILL DISPLAY "3 MIN DONE"
000222                          *   ON THE CONSOLE.  THE PROGRAM WILL NOT HALT, BUT RUN UNTIL MANUALLY
000223                          *   STOPPED.  TESTING FOR 3 MINUTES SHOULD BE SUFFICIENT.
000224                          *
000225                          *
000226                          *   THE FOLLOWING IS AN EXAMPLE OF THE END-OF-PASS REPORT
000227                          *
000228                          *    O.K.
000229                          *    CPFS4 PS    1
000230                          *    O.K.
000231                          *    CPFS4 PS    2
000232                          *    O.K.
000233                          *    CPFS4 PS    3
000234                          *        .
000235                          *        .
000236                          *        .
000237                          *        .
000238                          *    O.K.
000239                          *    CPFS4 PS    9
000240                          *    3 MIN DONE
000241                          *    O.K.
000242                          *    CPFS4 PS    10
000243                          *    O.K.
000244                          *    CPFS4 PS    11
000245                          *        .
000246                          *        .
000247                          *        .
000248                          *        .
000249                          *
000250                          *
000251                          *
000252                          *   ANY RESULT OTHER THAN THE ABOVE INDICATES A HARDWARE FAILURE AND THE
000253                          *   DIAGNOSTIC PROCEDURE FROM THE "LEVEL 6 SYSTEM CHECKOUT AND T&V MANUAL",
000254                          *   ORDER NO. "AW94", SHOULD BE FOLLOWED.
```

```
000255                          *
000256                          * ERROR REPORTING
000257                          * ----- ---------
000258                          *   DETECTED ERRORS ARE REPORTED AS FOLLOWS:
000259                          *
000260                          *     ERR CODE AT AAAA
000261                          *
000262                          *   WHERE:
000263                          *    CODE = ERROR CODE WITH UNIQUE FIRST FOUR CHARACTERS.
000264                          *    AAAA = LOCATION IN TEST IN PROGRAM FOR EXPLANATORY COMMENT.
000265                          *
000266                          *
000267                          *   THE ABOVE ERROR MESSAGE IS MINIMAL.  IT WILL BE SUPPLEMENTED,
000268                          *   DEPENDING ON THE ERROR, WITH ONE OR MORE OF THE FOLLOWING:
000269                          *
000270                          *
000271                          *    - REPLACE MB
000272                          *    - REPLACE DB 1
000273                          *    - REPLACE DB 2
000274                          *    - STATUS
000275                          *    - DATA IS, SHOULD BE
000276                          *    - LOCATION IN ERROR
000277                          *    - ZONE IN ERROR
000278                          *    - ERROR TYPE
000279                          *    - SPECIFIC ENGLISH SENTENCES
000280                          *              .
000281                          *              .
000282                          *              .
000283                          *              .
000284                          *
000285                          *   THERE ARE FAULTS WHICH WILL CAUSE FIRMWARE FREEZE, THUS,
000286                          *   PREVENTING THE DISPLAYING OF AN ERROR MESSAGE.  ANTICIPATING
000287                          *   THESE ERRORS, THE PROGRAM ISSUES WARNING MESSAGES AS SHOWN
000288                          *   BEFORE.
000289                          *
000290                          *
000291                          * NO CONSOLE
000292                          * -- -------
000293                          *   THIS PROGRAM CAN BE OPERATED WITHOUT A CONSOLE.  ERROR DATA IS LIMITED
000294                          *   TO "CODE" IN REGISTERS "R1" AND "R2" (I.E. "D1" AND "D2"), AND "AAAA"
000295                          *   IN REGISTER "B2".  COMMENT AT LOCATION "AAAA" WILL SPECIFY WHICH
000296                          *   REGISTERS CONTAIN ADDITIONAL INFORMATION.
000297                          *   ALL ENTRIES MUST BE MADE VIA REGISTER "R1"("D1").  THE PROCESS IS
000298                          *   FURTHER EXPLAINED IN MANUAL "Aw94" ENTITLED "LEVEL 6 SYSTEM CHECKOUT
000299                          *   AND T&V MANUAL".
000300                          *
000301                          *
000302                          *-------------------------------------------------------------------
000303                          *
000304                          *
```

```
000305                                  /
000306                                          CTRL  LINK ZV$TH          (ZV$THZ,ZV$TD)
000307                                          XLOC  ZV$AF
000308                                          XLOC  ZV$LR               MEMORY LOW
000309                                          XLOC  ZV$TTY              = 0 IF NO CONSOLE
000310                                          XLOC  ZV$ER
000311                                  *
000312                                          XLOC  ZHRTCI              RTC INITIAL VALUE - LOC HEX 0014
000313                                          XLOC  ZHRTCC              RTC CURRENT VALUE - LOC HEX 0015
000314                                          XLOC  ZHRTCL              RTC INTERRUPT LEVEL - LOC HEX 0016
000315                                          XLOC  ZHTH5               TRAP HANDLER POINTER NO. 5
000316                                          XLOC  ZHTH15              TRAP HANDLER POINTER NO. 15
000317                                          XLOC  ZHTH17              TRAP HANDLER POINTER NO. 17
000318                                  *
000319                                  *
000320  0000                            ZERO    EQU   $
000321  0100                                    ORG   ZERO+X'100'
000322                                  *
000323  0100  0F00 0000                START   NOP   <ZERO               SAF,LAF CHECK
000324  0102  0F00 0000                        NOP   <ZERO
000325  0104  8740 138A                        CL    PASCNT+$AF-1        CLEAR PASS COUNT
000326  0106  0F84                             B     >START1
000327                                  *
000328  0107  0F7F                     DONE    NOP   >$-1                3 MINUTES DONE
000329  0108  0F80 0135                        B     <NXTPAS
000330                                  *
000331                                  *
000332                                  *
000333                                  START1  CALL  ZV$RD,MSG1          INITIALIZE SYSTEM AND PRINT PROGRAM
        010A  FBC0 0003
        010C  D380 0000       X
        010E  0F80
        010F  1296
000334                                  *                                 NAME AND SYSTEM RESOURCES.
000335                                  *
000336                                  * ESTABLISH CHANNEL NUMBER, FUNCTION CODE TABLE
000337                                  *
000338  0110  9B80 14A1                ESTBL   LAB   $B1,<WCSTBL         ADRS OF TABLE
000339  0112  A870 FFF9                        LDR   $R2,=-7             7 WORDS
000340  0114  9800 1491                        LDR   $R1,<CPCHAN         **  ASSUME 0 FOR NOW  **
000341  0116  1006                             SOL   $R1,6               SHIFT CP CHAN NO TO BITS 8 AND 9
000342  0117  8B51                             LBC   =$R1,=Z'00C0'       COMPLEMENT BITS 8 AND 9
        0118  00C0
000343  0119  9AF1                ESTBL1  SRM   $R1,+$B1,=Z'00C0'   STORE UNDER MASK
        011A  00C0
000344  011B  27FE                             BINC  $R2,>ESTBL1
000345                                  *
000346                                  * INITIALIZE PASS COUNT
000347                                  *
000348  011C  CC80 1236                         LDB   $B4,<NULL           **  B4 RESERVED FOR PASS COUNT DISPLAY  **
000349                                  *
000350                                  * INITIALIZE REAL TIME CLOCK
000351                                  *
000352  011E  9870 FFFF                         LDR   $R1,=Z'FFFF'        MAX. VALUE
000353  0120  9F00 0000       X                 STR   $R1,<ZHRTCC         RTC CURRENT
000354  0122  9F00 0000       X                 STR   $R1,<ZHRTCI         RTC INITIAL
000355  0124  1C3F                             LDV   $R1,=63             LOWEST PRIORITY.  WON'T INTERRUPT.
000356  0125  9F00 0000       X                 STR   $R1,<ZHRTCL         RTC LEVEL
000357  0127  0004                             RTCN                      TURN ON CLOCK
000358                                  *
000359                                          CALL  ZV$TC,MSG1D         PRINT MESSAGE
        0128  FBC0 0003
        012A  D380 0000       X
        012C  0F80
        012D  12D8
000360                                          CALL  ZV$TC,MSG1E
        012E  FBC0 0003
        0130  D380 0000       X
        0132  0F80
        0133  12FE
000361  0134  0004                             RTCN
000362                                  *
000363                                  *
000364                                  *****
000365                                  *
000366                                  * COMES HERE FOR START OF NEXT PASS
000367                                  *
000368                                  *****
000369                                  *
000370                                  *
000371                                  *
000372                                  *
000373                                  *
000374                                  *
000375  0135  8070 8000                 NXTPAS  IO    =Z'8000',<OUTCON    INITIALIZE WCS
        0137  0000 14A1
000376  0139  0703                             BIOT  >INIT1
000377  013A  E380 114B                         LNJ   $B6,<NAKERR         **  ERROR "NAK"  **
000378                                  *                                 OUTPUT CONTROL SHOULD NEVER BE NAK'D.
000379                                  *                                 REPLACE WCS MOTHER BOARD.
000380                                  *
000381  013C  8054                      INIT1   IO    =$R4,<INSTAT        INPUT STATUS
        013D  0000 14A6
000382  013F  0704                             BIOT  >INIT2
000383  0140  E380 114B                         LNJ   $B6,<NAKERR         **  ERROR "STATUS"  **
000384  0142  0FF3                             B     >NXTPAS             INPUT STATUS SHOULD NEVER BE NAK'D.
000385                                  *                                 REPLACE WCS MOTHER BOARD.
000386                                  *
000387  0143  C970 0800                 INIT2   CMR   $R4,=Z'0800'        BIT 4 SET FOR NORMAL MODE?
000388  0145  098D                             BNE   >INIT3
000389  0146  CF00 148D                         STR   $R4,<MODBIT         YES - SAVE IT
000390  0148  89C0 1346                         CMZ   PASCNT+$AF-1        1ST PASS?
000391  014A  09A2                             BNE   >GETID              NO
000392                                          CALL  ZV$TC,MSG1B         PRINT "NORMAL MODE"
        014B  FBC0 0003
        014D  D380 0000       X
        014F  0F80
        0150  12AC
000393  0151  0F9B                             B     >GETID
000394                                  *
000395  0152  C970 0040                 INIT3   CMR   $R4,=Z'0040'        BIT 9 SET FOR TRANSPARENT MODE?
000396  0154  090C                             BE    >INIT5
000397                                  INIT4   CALL  ZV$ER,INIT4,MSG1A   **  ERROR "MODE"  **
```

```
          0155  FBC0 0003
          0157  D380 0000       X
          0159  0F80
          015A  0155
          015B  12A9
000398    015C  0004                    RTCN
000399                          *  ONLY BIT 4(NORMAL MODE) OR BIT 9(TRANSPARENT MODE) SHOULD BE SET.
000400                          *  STATUS READ IN R4.
000401                          *
000402    015D  E380 120C               LNJ   $B6,<PRSTS          PRINT STATUS
000403    015F  0FD6                    B     >NXTPAS             NO SENSE IN CONTINUING
000404                          *
000405    0160  CF00 148D       INIT5   STR   $R4,<MODBIT         SAVE TRANSPARENT BIT
000406    0162  89C0 132C               CMZ   PASCNT+$AF-1        1ST PASS?
000407    0164  0988                    BNE   >GETID              NO
000408                                  CALL  ZV$TC,MSG1C         PRINT "TRANSPARENT MODE"
          0165  FBC0 0003
          0167  D380 0000       X
          0169  0F80
          016A  12C2
000409    016B  0004                    RTCN
000410                          *
000411    016C  8054            GETID   IO    =$R4,<INID          INPUT ID
          016D  0000 14A7
000412    016F  0704                    BIOT  >GETID1
000413    0170  E380 114B               LNJ   $B6,<NAKERR         **  ERROR "NAK"  **
000414                          *                                 INPUT ID SHOULD NEVER BE NAK'D.
000415                          *                                 REPLACE WCS MOTHER BOARD.
000416                          *
000417    0172  0FAC                    B     >LET                GO TO NEXT TEST
000418                          *
000419    0173  C970 2610       GETID1  CMR   $R4,=Z'2610'
000420    0175  0929                    BE    >LET                O.K. - GO TO NEXT TEST.
000421                          *
000422                          GETID2  CALL  ZV$ER,GETID2,MSG8   **  ERROR "ID"  **
          0176  FBC0 0003
          0178  D380 0000       X
          017A  0F80
          017B  0176
          017C  1418
000423                          *                                 ID SHOULD BE HEX 2610.  ID READ IN R4.
000424                          *                                 REPLACE WCS MOTHER BOARD.
000425                          *
000426                                CALL  ZV$T,MSG6             PRINT "   IS"
          017D  FBC0 0003
          017F  D380 0000       X
          0181  0F80
          0182  1412
000427    0183  CF40 1303               STR   $R4,TMP1
000428                                  CALL  ZV$THZ,TMP1         PRINT ID READ
          0185  FBC0 0003
          0187  D380 0000       X
          0189  0F80
          018A  1487
000429                                CALL  ZV$T,MSG7             PRINT ",  SB"
          018B  FBC0 0003
          018D  D380 0000       X
          018F  0F80
          0190  1415
000430                                CALL  ZV$THZ,ID             PRINT "2610"
          0191  FBC0 0003
          0193  D380 0000       X
          0195  0F80
          0196  1492
000431                                CALL  ZV$T,MSG4             PRINT "   * REPLACE MB *"
          0197  FBC0 0003
          0199  D380 0000       X
          019B  0F80
          019C  1319
000432    019D  0004                    RTCN
000433                          *
000434                          *
000435                          *
000436                          *
000437                          **********************
000438                          *                    *
000439                          * LOADING ERROR TEST *
000440                          *                    *
000441                          **********************
000442                          *
000443                          * (STATUS BIT 10 - HEX 0020, FOR NATIVE PROM READ/WRITE)
000444                          *
000445                          *
000446    019E  8757            LET     CL    =$R7
000447    019F  8070 0000               IO    =Z'0000',<OUTADR    ADRS OF NATIVE PROM
          01A1  0000 14A2
000448    01A3  0703                    BIOT  >LET1
000449    01A4  E380 114B               LNJ   $B6,<NAKERR         **  ERROR "NAK"  **
000450                          *                                 OUTPUT ADRS SHOULD NEVER BE NAK'D.
000451                          *                                 REPLACE WCS MOTHER BOARD.
000452                          *
000453    01A6  8054            LET1    IO    =$R4,<INSTAT        INPUT STATUS
          01A7  0000 14A6
000454    01A9  0704                    BIOT  >LET2
000455    01AA  E380 114B               LNJ   $B6,<NAKERR         **  ERROR "NAK"  **
000456    01AC  0F87                    B     >LET3               INPUT STATUS SHOULD NEVER BE NAK'D.
000457                          *                                 REPLACE WCS MOTHER BOARD.
000458                          *
000459    01AD  D800 148D       LET2    LDR   $R5,<MODBIT         STATUS SB
000460    01AF  C955                    CMR   $R4,=$R5
000461    01B0  0903                    BE    >LET3
000462                          *
000463    01B1  E380 11B5               LNJ   $B6,<STERR          **  ERROR "STATUS"  **
000464                          *                                 ONLY MODE BIT SHOULD BE SET.
000465                          *                                 STATUS READ IN R4.
000466                          *                                 REPLACE WCS MOTHER BOARD.
000467                          *
000468    01B3  8057            LET3    IO    =$R7,<OUTDAT        OUTPUT DATA TO FORCE STATUS BIT 10
          01B4  0000 14A3
000469    01B6  0704                    BIOT  >LET4
000470    01B7  E380 114B               LNJ   $B6,<NAKERR         **  ERROR "NAK"  **
000471                          * SHOULD NOT GET NAK ERROR ON I/O DATA UNLESS A STATUS BIT (ERROR FLOP)
000472                          * WAS SET BY A PREVIOUS I/O DATA COMMAND.  REPLACE WCS MOTHER BOARD.
```

```
000473                                  *
000474  01B9  0F90                              B       >LET6
000475                                  *
000476  01BA  8054                      LET4    IO      =$R4,<INSTAT            INPUT STATUS
        01BB  0000  14A6
000477  01BD  0704                              BIOT    >LET5
000478  01BE  E380  114B                        LNJ     $B6,<NAKERR             **  ERROR "NAK"  **
000479  01C0  0F89                              B       >LET6                   INPUT STATUS SHOULD NEVER BE NAK'D.
000480                                  *                                       REPLACE WCS MOTHER BOARD.
000481                                  *
000482  01C1  D870  0020                LET5    LDR     $R5,=Z'0020'
000483  01C3  D400  148D                        OR      $R5,<MODBIT             R5=STATUS SB
000484  01C5  C955                              CMR     $R4,=$R5
000485  01C6  0903                              BE      >LET6
000486  01C7  E380  11B5                        LNJ     $B6,<STERR              **  ERROR "STATUS"  **
000487                                  *                                       SHOULD BE HEX 0020, BIT 10 FOR NATIVE
000488                                  *                                       PROM READ/WRITE, PLUS MODE BIT.
000489                                  *                                       STATUS READ IN R4.
000490                                  *                                       REPLACE WCS MOTHER BOARD.
000491                                  *
000492  01C9  8054                      LET6    IO      =$R4,<INSTAT            INPUT STATUS
        01CA  0000  14A6
000493  01CC  0704                              BIOT    >LET7
000494  01CD  E380  114B                        LNJ     $B6,<NAKERR             **  ERROR "NAK"  **
000495  01CF  0F87                              B       >LET8                   INPUT STATUS SHOULD NEVER BE NAK'D.
000496                                  *                                       REPLACE WCS MOTHER BOARD.
000497                                  *
000498  01D0  D800  148D                LET7    LDR     $R5,<MODBIT             STATUS SB
000499  01D2  C955                              CMR     $R4,=$R5
000500  01D3  0903                              BE      >LET8
000501  01D4  E380  11B5                        LNJ     $B6,<STERR              **  ERROR "STATUS" **
000502                                  *                                       BIT 10 SHOULD BE CLEARED FROM PREVIOUS
000503                                  *                                       INPUT STATUS COMMAND.  R4=STATUS READ.
000504                                  *                                       REPLACE WCS MOTHER BOARD.
000505                                  *
000506  01D6  8054                      LET8    IO      =$R4,<INDAT             INPUT DATA TO FORCE STATUS BIT 10 AGAIN.
        01D7  0000  14A5
000507  01D9  0705                              BIOT    >LET9
000508  01DA  E380  114B                        LNJ     $B6,<NAKERR             **  ERROR "NAK"  **
000509                                  *  SHOULD NOT GET NAK BECAUSE ERROR SHOULD HAVE BEEN CLEARED BY INPUT
000510                                  *  STATUS COMMAND ABOVE.  REPLACE WCS MOTHER BOARD.
000511                                  *
000512  01DC  0F81  003C                        B       NERT                    NO SENSE IN CONTINUING
000513                                  *                                       GO TO NEXT TEST.
000514                                  *
000515  01DE  1CF6                      LET9    LDV     $R1,=-10                FORCE 10 NAKS
000516  01DF  8057                      LET9A   IO      =$R7,<OUTDAT            OUTPUT DATA TO FORCE NAK
        01E0  0000  14A3
000517  01E2  0783                              BIOF    >LET10
000518  01E3  E380  1180                        LNJ     $B6,<ACKERR             **  ERROR "ACK"  **
000519                                  *  GOT ACK.  WCS SHOULD NAK BECAUSE ERROR FORCED BY ABOVE INPUT DATA
000520                                  *  COMMAND.  REPLACE WCS MOTHER BOARD.
000521                                  *
000522  01E5  17FA                      LET10   BINC    $R1,>LET9A              DO 10 TIMES
000523                                  *
000524  01E6  8054                              IO      =$R4,<INID              INPUT ID
        01E7  0000  14A7
000525  01E9  0704                              BIOT    >LET11
000526  01EA  E380  114B                        LNJ     $B6,<NAKERR             **  ERROR "NAK"  **
000527  01EC  0F91                              B       >LET13                  INPUT ID SHOULD NEVER BE NAK'D, EVEN
000528                                  *                                       IF ERROR FLOP IS SET.  REPLACE WCS
000529                                  *                                       MOTHER BOARD.
000530                                  *
000531  01ED  C970  2610                LET11   CMR     $R4,=Z'2610'
000532  01EF  090E                              BE      >LET13
000533                                  LET12   CALL    ZV$ER,LET12,MSG8        **  ERROR "ID"  **    ID READ IN R4.
        01F0  FBC0  0003
        01F2  D360  0000        X
        01F4  0F80
        01F5  01F0
        01F6  1418
000534                                  *                                       REPLACE WCS MOTHER BOARD.
000535                                  *
000536                                          CALL    ZV$T,MSG4               PRINT "   * REPLACE MB *"
        01F7  FBC0  0003
        01F9  D380  0000        X
        01FB  0F80
        01FC  1319
000537                                  *
000538  01FD  8054                      LET13   IO      =$R4,<INSTAT            INPUT STATUS
        01FE  0000  14A6
000539  0200  0704                              BIOT    >LET14
000540  0201  E380  114B                        LNJ     $B6,<NAKERR             **  ERROR "NAK"  **
000541  0203  0F89                              B       >LET15                  INPUT STATUS SHOULD NEVER BE NAK'D.
000542                                  *                                       REPLACE WCS MOTHER BOARD.
000543                                  *
000544  0204  D870  0020                LET14   LDR     $R5,=Z'0020'
000545  0206  D400  148D                        OR      $R5,<MODBIT             R5=STATUS SB
000546  0208  C955                              CMR     $R4,=$R5
000547  0209  0903                              BE      >LET15
000548  020A  E380  11B5                        LNJ     $B6,<STERR              **  ERROR "STATUS"  **
000549                                  *                                       SHOULD BE HEX 0020, BIT 10 FOR NATIVE
000550                                  *                                       PROM READ/WRITE, PLUS MODE BIT.
000551                                  *                                       R4=STATUS READ.
000552                                  *                                       REPLACE WCS MOTHER BOARD.
000553                                  *
000554  020C  8054                      LET15   IO      =$R4,<INSTAT            INPUT STATUS.
        020D  0000  14A6
000555  020F  0704                              BIOT    >LET16
000556  0210  E380  114B                        LNJ     $B6,<NAKERR             **  ERROR "NAK"  **
000557                                  *                                       INPUT STATUS SHOULD NEVER BE NAK'D.
000558                                  *                                       REPLACE WCS MOTHER BOARD.
000559                                  *
000560  0212  0F87                              B       >NERT                   GO TO NEXT TEST.
000561                                  *
000562  0213  D800  148D                LET16   LDR     $R5,<MODBIT             STATUS SB
000563  0215  D954                              CMR     $R5,=$R4
000564  0216  0903                              BE      >NERT                   IF O.K., GO TO NEXT TEST
000565                                  *
000566  0217  E380  11B5                        LNJ     $B6,<STERR              **  ERROR "STATUS"  **
000567                                  *                                       ONLY MODE BIT SHOULD BE SET.
000568                                  *                                       STATUS READ IN R4.
000569                                  *                                       REPLACE WCS MOTHER BOARD.
```

```
000570                          *
000571                          *
000572                          *
000573                          *
000574                          ******************************
000575                          *                            *
000576                          * NON-EXISTANT RESOURCE TEST *
000577                          *                            *
000578                          ******************************
000579                          *
000580                          * (STATUS BIT 13 - HEX 0004, FOR NON-EXISTANT RESOURCE READ/WRITE)
000581                          *
000582                          *
000583  0219  8757              NERT    CL      =$R7
000584  021A  8070 0400                 IO      =Z'0400',<OUTADR        ADRS OF AREA NOT AVAILABLE FOR PROMS
      021C  0000 14A2
000585                          *                                       OR RAMS.
000586  021E  0703                      BIOT    >NERT1
000587  021F  E380 114B                 LNJ     $B6,<NAKERR             ** ERROR "NAK" **
000588                          *                                       OUTPUT ADRS SHOULD NEVER BE NAK'D.
000589                          *                                       REPLACE WCS MOTHER BOARD.
000590                          *
000591  0221  8054              NERT1   IO      =$R4,<INSTAT            INPUT STATUS
      0222  0000 14A6
000592  0224  0704                      BIOT    >NERT2
000593  0225  E380 114B                 LNJ     $B6,<NAKERR             ** ERROR "NAK" **
000594  0227  0F87                      B       >NERT3                  INPUT STATUS SHOULD NEVER BE NAK'D.
000595                          *                                       REPLACE WCS MOTHER BOARD.
000596                          *
000597  0228  D800 148D          NERT2   LDR     $R5,<MODBIT             STATUS 5B
000598  022A  C955                      CMR     $R4,=$R5
000599  022B  0903                      BE      >NERT3
000600  022C  E380 11B5                 LNJ     $B6,<STERR              ** ERROR "STATUS" **
000601                          *                                       ONLY MODE BIT SHOULD BE SET.
000602                          *                                       STATUS READ IN R4.
000603                          *                                       REPLACE WCS MOTHER BOARD.
000604                          *
000605  022E  8054              NERT3   IO      =$R4,<INDA1             INPUT DATA TO FORCE STATUS BIT 13
      022F  0000 14A5
000606  0231  0704                      BIOT    >NERT4
000607  0232  E380 114B                 LNJ     $B6,<NAKERR             ** ERROR "NAK" **
000608                          *  SHOULD NOT GET NAK ON I/O DATA UNLESS A STATUS BIT (ERROR FLOP) WAS
000609                          *  SET BY A PREVIOUS I/O DATA COMMAND.  REPLACE WCS MOTHER BOARD.
000610                          *
000611  0234  0F90                      B       >NERT6
000612                          *
000613  0235  8054              NERT4   IO      =$R4,<INSTAT            INPUT STATUS
      0236  0000 14A6
000614  0238  0704                      BIOT    >NERT5
000615  0239  E380 114B                 LNJ     $B6,<NAKERR             ** ERROR "NAK" **
000616  023B  0F89                      B       >NERT6                  INPUT STATUS SHOULD NEVER BE NAK'D.
000617                          *                                       REPLACE WCS MOTHER BOARD.
000618                          *
000619  023C  D870 0004          NERT5   LDR     $R5,=Z'0004'
000620  023E  D400 148D                 OR      $R5,<MODBIT             R5=STATUS 5B
000621  0240  C955                      CMR     $R4,=$R5
000622  0241  0903                      BE      >NERT6
000623  0242  E380 11B5                 LNJ     $B6,<STERR              ** ERROR "STATUS" **
000624                          *                                       SHOULD BE HEX 0004, BIT 13 FOR NON-
000625                          *                                       EXISTANT RESOURCE READ/WRITE,
000626                          *                                       PLUS MODE BIT.
000627                          *                                       STATUS READ IN R4.
000628                          *                                       REPLACE WCS MOTHER BOARD.
000629                          *
000630  0244  8054              NERT6   IO      =$R4,<INSTAT            INPUT STATUS
      0245  0000 14A6
000631  0247  0704                      BIOT    >NERT7
000632  0248  E380 114B                 LNJ     $B6,<NAKERR             ** ERROR "NAK" **
000633  024A  0F87                      B       >NERT8                  INPUT STATUS SHOULD NEVER BE NAK'D.
000634                          *                                       REPLACE WCS MOTHER BOARD.
000635                          *
000636  024B  D800 148D          NERT7   LDR     $R5,<MODBIT             STATUS 5B
000637  024D  C955                      CMR     $R4,=$R5
000638  024E  0903                      BE      >NERT8
000639  024F  E380 11B5                 LNJ     $B6,<STERR              ** ERROR "STATUS" **
000640                          *                                       BIT 13 SHOULD BE CLEARED FROM PREVIOUS
000641                          *                                       INPUT STATUS COMMAND.  R4=STATUS READ.
000642                          *                                       REPLACE WCS MOTHER BOARD.
000643                          *
000644  0251  8057              NERT8   IO      =$R7,<OUTDAT            OUTPUT DATA TO FORCE STATUS BIT 13 AGAIN
      0252  0000 14A3
000645  0254  0705                      BIOT    >NERT9
000646  0255  E380 114B                 LNJ     $B6,<NAKERR             ** ERROR "NAK" **
000647                          *  SHOULD NOT GET NAK BECAUSE ERROR SHOULD HAVE BEEN CLEARED BY INPUT
000648                          *  STATUS COMMAND ABOVE.  REPLACE WCS MOTHER BOARD.
000649                          *
000650  0257  0F81 003C                  B       DNR                     NO SENSE IN CONTINUING
000651                          *                                       GO TO NEXT TEST
000652                          *
000653  0259  1CF6              NERT9   LDV     $R1,=-10                FORCE 10 NAKS
000654  025A  8054              NERT9A  IO      =$R4,<INDAT             INPUT DATA TO FORCE NAK.
      025B  0000 14A5
000655  025D  0783                      BIOF    >NERT10
000656  025E  E380 1180                 LNJ     $B6,<ACKERR             ** ERROR "ACK" **
000657                          *  GOT ACK.  WCS SHOULD NAK BECAUSE ERROR FORCED BY ABOVE OUTPUT DATA
000658                          *  COMMAND.  REPLACE WCS MOTHER BOARD.
000659                          *
000660  0260  17FA              NERT10  BINC    $R1,>NERT9A             DO 10 TIMES
000661                          *
000662  0261  8054                      IO      =$R4,<INID              INPUT ID
      0262  0000 14A7
000663  0264  0704                      BIOT    >NERT11
000664  0265  E380 114B                 LNJ     $B6,<NAKERR             ** ERROR "NAK" **
000665  0267  0F91                      B       >NERT13                 INPUT ID SHOULD NEVER BE NAK'D, EVEN
000666                          *                                       IF ERROR FLOP IS SET.
000667                          *                                       REPLACE WCS MOTHER BOARD.
000668                          *
000669  0268  C970 2610          NERT11  CMR     $R4,=Z'2610'
000670  026A  090E                      BE      >NERT13
000671                          NERT12  CALL    ZV$ER,NERT12,MSGB       ** ERROR "ID" **    ID READ IN R4.
      026B  FBC0 0003
      026D  0380 0000       X
      026F  0F80
```

```
                0270  026B
                0271  1418
000672                                  *                                       REPLACE WCS MOTHER BOARD.
000673                                  *
000674                                          CALL  ZV$T,MSG4                 PRINT "   * REPLACE MB *"
                0272  FBC0 0003
                0274  D380 0000        X
                0276  0F80
                0277  1319
000675                                  *
000676  0278  8054                     NERT13 IO    =$R4,<INSTAT               INPUT STATUS
              0279  0000 14A6
000677  027B  0704                             BIOT  >NERT14
000678  027C  E380 114B                         LNJ   $B6,<NAKERR               **  ERROR "NAK"  **
000679  027E  0F89                             B     >NERT15                   INPUT STATUS SHOULD NEVER BE NAK'D.
000680                                  *                                       REPLACE WCS MOTHER BOARD.
000681                                  *
000682  027F  D870 0004                 NERT14 LDR   $R5,=Z'0004'
000683  0281  D400 148D                         OR    $R5,<MODBIT               R5=STATUS 5B
000684  0283  C955                             CMR   $R4,=$R5
000685  0284  0903                             BE    >NERT15
000686  0285  E380 11B5                         LNJ   $B6,<STERR                **  ERROR "STATUS"  **
000687                                  *                                       SHOULD BE HEX 0004, BIT 13 FOR NON-
000688                                  *                                       EXISTANT RESOURCE READ/WRITE,
000689                                  *                                       PLUS MODE BIT.
000690                                  *                                       STATUS READ IN R4.
000691                                  *                                       REPLACE WCS MOTHER BOARD.
000692                                  *
000693  0287  8054                     NERT15 IO    =$R4,<INSTAT               INPUT STATUS
              0288  0000 14A6
000694  028A  0704                             BIOT  >NERT16
000695  028B  E380 114B                         LNJ   $B6,<NAKERR               **  ERROR "NAK"  **
000696                                  *                                       INPUT STATUS SHOULD NEVER BE NAK'D.
000697                                  *                                       REPLACE WCS MOTHER BOARD.
000698                                  *
000699  028D  0F87                             B     >DNR                      GO TO NEXT TEST.
000700                                  *
000701  028E  D800 148D                 NERT16 LDR   $R5,<MODBIT               STATUS 5B
000702  0290  C955                             CMR   $R4,=$R5
000703  0291  0903                             BE    >DNR                      IF O.K., GO TO NEXT TEST
000704                                  *
000705  0292  E380 11B5                         LNJ   $B6,<STERR                **  ERROR "STATUS"  **
000706                                  *                                       ONLY MODE BIT SHOULD BE SET.
000707                                  *                                       STATUS READ IN R4.
000708                                  *                                       REPLACE WCS MOTHER BOARD.
000709                                  *
000710                                  *
000711                                  *
000712                                  *
000713                                  ***********************************
000714                                  *                                 *
000715                                  * DETERMINE NO. RAMS (AND REPORT) *
000716                                  *                                 *
000717                                  ***********************************
000718                                  *
000719                                  * (ALSO VERIFY BOUNDARY ERRORS)
000720                                  *
000721                                  *
000722  0294  8757                     DNR    CL    =$R7
000723  0295  8700 1494                        CL    <DNRERF                   CLEAR FLAGS
000724  0297  8700 1495                        CL    <RAM1FL
000725  0299  8700 1496                        CL    <RAM2FL
000726                                  *
000727                                  * DETERMINE IF RAM NO.1 PRESENT
000728                                  *
000729  029B  8070 0800                         IO    =Z'0800',<OUTADR          ADRS OF 1ST RAM
              029D  0000 14A2
000730  029F  0705                             BIOT  >DNR1
000731  02A0  E380 114B                         LNJ   $B6,<NAKERR               **  ERROR "NAK"  **
000732  02A2  0F80 02FB                         B     <DNR10                    OUTPUT ADDRESS SHOULD NEVER BE NAK'D.
000733                                  *                                       REPLACE WCS MOTHER BOARD.
000734                                  *
000735  02A4  8057                     DNR1   IO    =$R7,<OUTDAT              OUTPUT DATA - WILL SET STATUS BIT 13,
              02A5  0000 14A3
000736  02A7  0705                             BIOT  >DNR2                     IF RAM 1 NOT PRESENT.
000737                                  *
000738  02A8  E380 114B                         LNJ   $B6,<NAKERR               **  ERROR "NAK"  **
000739  02AA  0F80 02FB                         B     <DNR10                    REPLACE WCS MOTHER BOARD.
000740                                  *
000741  02AC  8057                     DNR2   IO    =$R7,<OUTDAT              SHOULD NAK IF RAM 1 NOT PRESENT
              02AD  0000 14A3
000742  02AF  07A8                             BIOF  >DNR6
000743                                  *
000744                                  * GOT ACK - LOOKS LIKE RAM NO.1 MIGHT BE PRESENT
000745                                  *
000746  02B0  8054                             IO    =$R4,<INSTAT
              02B1  0000 14A6
000747  02B3  0705                             BIOT  >DNR3
000748  02B4  E380 114B                         LNJ   $B6,<NAKERR               **  ERROR "NAK"  **
000749  02B6  0F81 0044                         B     DNR10                     INPUT STATUS SHOULD NEVER BE NAK'D.
000750                                  *                                       REPLACE WCS MOTHER BOARD.
000751                                  *
000752  02B8  82D4                     DNR3   LB    =$R4,=Z'0004'             FETCH STATUS BIT 13
              02B9  0004
000753  02BA  0587                             BBF   >DNR4                     5B 0
000754  02BB  D800 148D                         LDR   $R5,<MODBIT               STATUS 5B
000755                                  *
000756  02BD  E380 11B5                         LNJ   $B6,<STERR                **  ERROR "STATUS"  **
000757  02BF  0F81 003B                         B     DNR10                     BIT 13 5B 0 BECAUSE NO NAK ON 2ND
000758                                  *                                       OUTPUT DATA COMMAND ABOVE.
000759                                  *                                       STATUS READ IN R4.
000760                                  *                                       REPLACE WCS MOTHER BOARD.
000761                                  *
000762  02C1  D800 148D                 DNR4   LDR   $R5,<MODBIT               STATUS 5B
000763  02C3  C955                             CMR   $R4,=$R5
000764  02C4  0904                             BE    >DNR5
000765  02C5  E380 11B5                         LNJ   $B6,<STERR                **  ERROR "STATUS"  **
000766  02C7  0FB4                             B     >DNR10                    STATUS 5B 0 BECAUSE NO NAK ON 2ND
000767                                  *                                       OUTPUT DATA COMMAND ABOVE.
000768                                  *                                       STATUS READ IN R4.
000769                                  *                                       REPLACE WCS MOTHER BOARD.
000770                                  *
000771  02C8  1C01                     DNR5   LDV   $R1,=1                     SHOW RAM 1 PRESENT
```

```
000772  02C9  9F00 1495          STR   $R1,<RAM1FL
000773  02CB  89C0 11C3          CMZ   PASCNT+$AF-1        1ST PASS?
000774  02CD  0981 0039          BNE   DNR11               NO
000775                           CALL  ZV$TC,MSG12         PRINT "RAM NO.1 PRESENT"
        02CF  FBC0 0003
        02D1  D380 0000      X
        02D3  0F80
        02D4  142B
000776  02D5  0004               RTCN
000777  02D6  0FB1               B     >DNR11
000778                       *
000779                       * GOT NAK - LOOKS LIKE RAM NO.1 MIGHT NOT BE PRESENT
000780                       *
000781  02D7  8054           DNR6  IO    =$R4,<INSTAT
        02D8  0000 14A6
000782  02DA  0704               BIOT  >DNR7
000783  02DB  E380 114B          LNJ   $B6,<NAKERR         **  ERROR "NAK"  **
000784  02DD  0F9E               B     >DNR10              INPUT STATUS SHOULD NEVER BE NAK'D.
000785                       *                            REPLACE WCS MOTHER BOARD.
000786                       *
000787  02DE  82D4           DNR7  LB    =$R4,=Z'0004'     FETCH STATUS BIT 13.
        02DF  0004
000788  02E0  0506               BBT   >DNR8               SHOULD BE SET.
000789  02E1  D870 FFFF          LDR   $R5,=Z'FFFF'        DON'T PRINT STATUS SB
000790                       *
000791  02E3  E380 11B5          LNJ   $B6,<STERR          **  ERROR "STATUS"  **
000792  02E5  0F96               B     >DNR10              NAK NOT CAUSED BY STATUS BIT 13.
000793                       *                            STATUS READ IN R4.
000794                       *                            REPLACE WCS MOTHER BOARD.
000795                       *
000796  02E6  5C04           DNR8  LDV   $R5,=4
000797  02E7  D400 148D          OR    $R5,<MODBIT         R5=STATUS SB
000798  02E9  C955               CMR   $R4,=$R5
000799  02EA  0906               BE    >DNR9
000800  02EB  D870 FFFF          LDR   $R5,=Z'FFFF'        DON'T PRINT STATUS SB
000801                       *
000802  02ED  E380 11B5          LNJ   $B6,<STERR          **  ERROR "STATUS"  **
000803  02EF  0F8C               B     >DNR10              STATUS READ IN R4.
000804                       *                            REPLACE WCS MOTHER BOARD.
000805                       *
000806  02F0  89C0 119E      DNR9  CMZ   PASCNT+$AF-1        1ST PASS?
000807  02F2  0995               BNE   >DNR11              NO
000808                           CALL  ZV$TC,MSG13         PRINT "RAM NO.1 NOT PRESENT"
        02F3  FBC0 0003
        02F5  D380 0000      X
        02F7  0F80
        02F8  1434
000809  02F9  0004               RTCN
000810  02FA  0F8D               B     >DNR11
000811                       *
000812                       *
000813  02FB  1C01           DNR10 LDV   $R1,=1              SHOW GOT ERROR DURING RAM 1
000814  02FC  9F00 1494          STR   $R1,<DNRERF         VERIFICATION.  DON'T KNOW IF PRESENT
000815                       *                            OR NOT.  CHECK RAM NO.2.
000816                       *
000817  02FE  8070 8000          IO    =Z'8000',<OUTCON    INITIALIZE WCS
        0300  0000 14A1
000818  0302  0705               BIOT  >DNR11
000819  0303  E380 114B          LNJ   $B6,<NAKERR         **  ERROR "NAK"  **
000820                       *                            OUTPUT CONTROL SHOULD NEVER BE NAK'D.
000821                       *                            REPLACE WCS MOTHER BOARD.
000822                       *
000823  0305  0F80 0135          B     <NXTPAS             NO SENSE IN CONTINUING TEST.
000824                       *
000825                       *
000826                       * DETERMINE IF RAM NO.2 PRESENT
000827                       *
000828  0307  8070 0C00      DNR11 IO    =Z'0C00',<OUTADR  ADRS OF 2ND RAM
        0309  0000 14A2
000829  030B  0705               BIOT  >DNR12
000830  030C  E380 114B          LNJ   $B6,<NAKERR         **  ERROR "NAK"  **
000831                       *                            OUTPUT ADRS SHOULD NEVER BE NAK'D.
000832                       *                            REPLACE WCS MOTHER BOARD.
000833                       *
000834  030E  0F80 0135          B     <NXTPAS             NO SENSE IN CONTINUING TEST.
000835                       *
000836  0310  8054           DNR12 IO    =$R4,<INDAT       INPUT DATA - WILL SET STATUS BIT 13
        0311  0000 14A5
000837  0313  0703               BIOT  >DNR13              IF RAM 1 NOT PRESENT
000838                       *
000839  0314  E380 114B          LNJ   $B6,<NAKERR         **  ERROR "NAK"  **
000840                       *                            REPLACE WCS MOTHER BOARD.
000841                       *
000842  0316  8054           DNR13 IO    =$R4,<INDAT       SHOULD NAK IF RAM 2 NOT PRESENT
        0317  0000 14A5
000843  0319  07A8               BIOF  >DNR17
000844                       *
000845                       * GOT ACK - LOOKS LIKE RAM NO.2 MIGHT BE PRESENT
000846                       *
000847  031A  8054               IO    =$R4,<INSTAT
        031B  0000 14A6
000848  031D  0705               BIOT  >DNR14
000849  031E  E380 114B          LNJ   $B6,<NAKERR         **  ERROR "NAK"  **
000850                       *                            INPUT STATUS SHOULD NEVER BE NAK'D.
000851                       *                            REPLACE WCS MOTHER BOARD.
000852                       *
000853  0320  0F80 0135          B     <NXTPAS             NO SENSE IN CONTINUING TEST.
000854                       *
000855  0322  82D4           DNR14 LB    =$R4,=Z'0004'     FETCH STATUS BIT 13.
        0323  0004
000856  0324  0587               BBF   >DNR15              SB 0.
000857  0325  D800 148D          LDR   $R5,<MODBIT         STATUS SB
000858                       *
000859  0327  E380 11B5          LNJ   $B6,<STERR          **  ERROR "STATUS"  **
000860                       *                            BIT 13 SB 0 BECAUSE NO NAK ON 2ND
000861                       *                            INPUT DATA COMMAND ABOVE.
000862                       *                            STATUS READ IN R4.
000863                       *                            REPLACE WCS MOTHER BOARD.
000864                       *
000865  0329  0F80 0135          B     <NXTPAS             NO SENSE IN CONTINUING TEST.
000866                       *
000867  032B  D800 148D      DNR15 LDR   $R5,<MODBIT         STATUS SB
000868  032D  C955               CMR   $R4,=$R5
```

```
000869  032E  0905                        BE    >DNR16
000870  032F  E380 11B5                   LNJ   $B6,<STERR          **  ERROR "STATUS"  **
000871                              *                               STATUS SB 0 BECAUSE NO NAK ON 2ND
000872                              *                               INPUT DATA COMMAND ABOVE.
000873                              *                               STATUS READ IN R4.
000874                              *                               REPLACE WCS MOTHER BOARD.
000875                              *
000876  0331  0F80 0135                    B     <NXTPAS             NO SENSE IN CONTINUING TEST.
000877                              *
000878  0333  1C01                  DNR16 LDV   $R1,=1              SHOW RAM NO.2 PRESENT
000879  0334  9F00 1496                    STR   $R1,<RAM2FL
000880  0336  89C0 1158                    CMZ   PASCNT+$AF-1        1ST PASS?
000881  0338  09AF                        BNE   >DNR21              NO
000882                                    CALL  ZV$TC,MSG14         PRINT "RAM NO.2 PRESENT"
        0339  FBC0 0003
        033B  D380 0000       X
        033D  0F80
        033E  143F
000883  033F  0004
000884  0340  0FA7                        B     >DNR21
000885                              *
000886                              * GOT NAK - LOOKS LIKE RAM NO.2 MIGHT NOT BE PRESENT
000887                              *
000888  0341  8054                  DNR17 IO    =$R4,<INSTAT
        0342  0000 14A6
000889  0344  0705                        BIOT  >DNR18
000890  0345  E380 114B                   LNJ   $B6,<NAKERR         **  ERROR "NAK"  **
000891                              *                               INPUT STATUS SHOULD NEVER BE NAK'D.
000892                              *                               REPLACE WCS MOTHER BOARD.
000893                              *
000894  0347  0F80 0135                    B     <NXTPAS             NO SENSE IN CONTINUING TEST.
000895                              *
000896  0349  82D4                  DNR18 LB    =$R4,=Z'0004'       FETCH STATUS BIT 13.
        034A  0004
000897  034B  0507                        BBT   >DNR19              SHOULD BE SET.
000898  034C  D870 FFFF                    LDR   $R5,=Z'FFFF'        DON'T PRINT STATUS SB
000899                              *
000900  034E  E380 11B5                   LNJ   $B6,<STERR          **  ERROR "STATUS"  **
000901                              *                               NAK NOT CAUSED BY STATUS BIT 13.
000902                              *                               STATUS READ IN R4.
000903                              *                               REPLACE WCS MOTHER BOARD.
000904                              *
000905  0350  0F80 0135                    B     <NXTPAS             NO SENSE IN CONTINUING TEST.
000906                              *
000907  0352  5C04                  DNR19 LDV   $R5,=4
000908  0353  D400 148D                    OR    $R5,<MODBIT         R5=STATUS SB
000909  0355  C955                        CMR   $R4,=$R5
000910  0356  0907                        BE    >DNR20
000911  0357  D870 FFFF                    LDR   $R5,=Z'FFFF'        DON'T PRINT STATUS SB
000912                              *
000913  0359  E380 11B5                   LNJ   $B6,<STERR          **  ERROR "STATUS"  **
000914                              *                               STATUS READ IN R4.
000915                              *                               REPLACE WCS MOTHER BOARD.
000916                              *
000917  035B  0F80 0135                    B     <NXTPAS             NO SENSE IN CONTINUING TEST.
000918                              *
000919  035D  89C0 1131              DNR20 CMZ   PASCNT+$AF-1        1ST PASS?
000920  035F  0988                        BNE   >DNR21              NO
000921                                    CALL  ZV$TC,MSG15         PRINT "RAM NO.2 NOT PRESENT"
        0360  FBC0 0003
        0362  D380 0000       X
        0364  0F80
        0365  1448
000922  0366  0004                        RTCN
000923                              *
000924                              *
000925  0367  8980 1494              DNR21 CMZ   <DNRERF
000926  0369  0980 0135                    BNE   <NXTPAS             NO SENSE IN CONTINUING TEST.  CAN'T
000927                              *                               DETERMINE IF RAM NO.1 PRESENT OR NOT.
000928                              *
000929                              *
000930  036B  8980 1495                    CMZ   <RAM1FL
000931  036D  0900 03C1                    BE    <DNR31              NO RAM 1.  CHECK IF RAM 2 PRESENT.
000932  036F  8980 1496                    CMZ   <RAM2FL
000933  0371  0980 03C5                    BNE   <DNR32              RAM 1 AND RAM 2 BOTH PRESENT.
000934                              *
000935                              * RAM NO.1 PRESENT, RAM NO.2 NOT PRESENT.  VERIFY STATUS BIT 13 (NON-
000936                              * EXISTANT RESOURCE) AND NAK WHEN OVERSTEP RAM 1'S BOUNDARY.
000937                              *
000938  0373  8070 0BFF                DNR22 IO    =Z'0BFF',<OUTADR      ADRS OF LAST 64-BIT WORD OF RAM 1
        0375  0000 14A2
000939  0377  0705                        BIOT  >DNR22
000940  0378  E380 114B                   LNJ   $B6,<NAKERR         **  ERROR "NAK"  **
000941                              *                               OUTPUT ADRS SHOULD NEVER BE NAK'D.
000942                              *                               REPLACE WCS MOTHER BOARD.
000943                              *
000944  037A  0F80 040A                    B     <PET                GO TO NEXT TEST
000945                              *
000946  037C  8057                  DNR22 IO    =$R7,<OUTDAT        I/O 1
        037D  0000 14A3
000947  037F  0705                        BIOT  >DNR23
000948  0380  E380 114B                   LNJ   $B6,<NAKERR         **  ERROR "NAK"  **   REPLACE WCS MB
000949                              *
000950  0382  0F80 040A                    B     <PET                GO TO NEXT TEST
000951                              *
000952  0384  8054                  DNR23 IO    =$R4,<INDAT         I/O 2
        0385  0000 14A5
000953  0387  0705                        BIOT  >DNR24
000954  0388  E380 114B                   LNJ   $B6,<NAKERR         **  ERROR "NAK"  **   REPLACE WCS MB
000955                              *
000956  038A  0F80 040A                    B     <PET                GO TO NEXT TEST
000957                              *
000958  038C  8057                  DNR24 IO    =$R7,<OUTDAT        I/O 3
        038D  0000 14A3
000959  038F  0705                        BIOT  >DNR25
000960  0390  E380 114B                   LNJ   $B6,<NAKERR         **  ERROR "NAK"  **   REPLACE WCS MB
000961                              *
000962  0392  0F80 040A                    B     <PET                GO TO NEXT TEST
000963                              *
000964  0394  8054                  DNR25 IO    =$R4,<INDAT         I/O 4
        0395  0000 14A5
000965  0397  0705                        BIOT  >DNR26
000966  0398  E380 114B                   LNJ   $B6,<NAKERR         **  ERROR "NAK"  **   REPLACE WCS MB
```

```
000967                                  *
000968  039A  0F80 040A                         B     <PET                    GO TO NEXT TEST
000969                                  *
000970  039C  8057                      DNR26  IO    =$R7,<OUTDAT            I/O 5 - SHOULD FORCE STATUS BIT 13
        039D  0000 14A3
000971  039F  0705                             BIOT  >DNR27
000972  03A0  E380 114B                         LNJ   $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
000973                                  *
000974  03A2  0F80 040A                         B     <PET                    GO TO NEXT TEST
000975                                  *
000976  03A4  1CF6                      DNR27  LDV   $R1,=-10                FORCE 10 NAKS
000977  03A5  8054                      DNR28  IO    =$R4,<INDAT             I/O 6-15 - SHOULD CAUSE NAKS
        03A6  0000 14A5
000978  03A8  0785                             BIOF  >DNR29
000979  03A9  E380 1180                         LNJ   $B6,<ACKERR             **  ERROR "ACK"  **
000980                                  *  GOT ACK.  SHOULD HAVE BEEN NAK'D BECAUSE BIT 13 SHOULD BE SET.
000981                                  *  REPLACE WCS MOTHER BOARD.
000982                                  *
000983  03AB  0F80 040A                         B     <PET                    GO TO NEXT TEST
000984                                  *
000985  03AD  17F8                      DNR29  BINC  $R1,>DNR28              DO 10 TIMES
000986                                  *
000987  03AE  8054                             IO    =$R4,<INSTAT
        03AF  0000 14A6
000988  03B1  0705                             BIOT  >DNR30
000989  03B2  E380 114B                         LNJ   $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
000990                                  *                                     INPUT STATUS SHOULD NEVER BE NAK'D.
000991                                  *
000992  03B4  0F80 040A                         B     <PET                    GO TO NEXT TEST
000993                                  *
000994  03B6  D870 0004                 DNR30  LDR   $R5,=Z'0004'
000995  03B8  D400 148D                         OR    $R5,<MODBIT             R5=STATUS S6
000996  03BA  D954                             CMR   $R5,=$R4
000997  03BB  0900 040A                         BE    <PET                    IF O.K., GO TO NEXT TEST
000998                                  *
000999  03BD  E380 11B5                         LNJ   $B6,<STERR              **  ERROR "STATUS"  **
001000                                  *  NAK SHOULD HAVE BEEN CAUSED BY STATUS BIT 13 (NON-EXISTANT RESOURCE),
001001                                  *  AND ONLY BIT 13.  STATUS SHOULD BE HEX 0004, PLUS MODE BIT.
001002                                  *  STATUS READ IN R4.  REPLACE WCS MOTHER BOARD.
001003                                  *
001004  03BF  0F80 040A                         B     <PET                    GO TO NEXT TEST
001005                                  *
001006                                  * RAM 1 NOT PRESENT.  CHECK IF RAM 2 PRESENT.
001007                                  *
001008  03C1  8980 1496                 DNR31  CMZ   <RAM2FL
001009  03C3  0900 0135                         BE    <NXTPAS                 NO RAMS DETECTED.  NO SENSE IN
001010                                  *                                     CONTINUING TEST.
001011                                  *
001012                                  * RAM 2 PRESENT (RAM 1 N/A).  VERIFY STATUS BIT 10 (NATIVE PROM LOADING
001013                                  * ERROR) AND NAK WHEN OVERSTEP RAM 2'S BOUNDARY.
001014                                  *
001015  03C5  8070 0FFF                 DNR32  IO    =Z'0FFF',<OUTADR        ADRS OF LAST 64-BIT WORD OF RAM 2
        03C7  0000 14A2
001016  03C9  0705                             BIOT  >DNR33
001017  03CA  E380 114B                         LNJ   $B6,<NAKERR             **  ERROR "NAK"  **
001018                                  *                                     OUTPUT ADRS SHOULD NEVER BE NAK'D.
001019                                  *                                     REPLACE WCS MOTHER BOARD.
001020                                  *
001021  03CC  0F81 003D                         B     PET                     GO TO NEXT TEST
001022                                  *
001023  03CE  8054                      DNR33  IO    =$R4,<INDAT             I/O 1
        03CF  0000 14A5
001024  03D1  0705                             BIOT  >DNR34
001025  03D2  E380 114B                         LNJ   $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001026                                  *
001027  03D4  0F81 0035                         B     PET                     GO TO NEXT TEST
001028                                  *
001029  03D6  8057                      DNR34  IO    =$R7,<OUTDAT            I/O 2
        03D7  0000 14A3
001030  03D9  0704                             BIOT  >DNR35
001031  03DA  E380 114B                         LNJ   $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001032                                  *
001033  03DC  0FAE                              B     >PET                    GO TO NEXT TEST
001034                                  *
001035  03DD  8054                      DNR35  IO    =$R4,<INDAT             I/O 3
        03DE  0000 14A5
001036  03E0  0704                             BIOT  >DNR36
001037  03E1  E380 114B                         LNJ   $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001038                                  *
001039  03E3  0FA7                              B     >PET                    GO TO NEXT TEST
001040                                  *
001041  03E4  8057                      DNR36  IO    =$R7,<OUTDAT            I/O 4
        03E5  0000 14A3
001042  03E7  0704                             BIOT  >DNR37
001043  03E8  E380 114B                         LNJ   $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001044                                  *
001045  03EA  0FA0                              B     >PET                    GO TO NEXT TEST
001046                                  *
001047  03EB  8054                      DNR37  IO    =$R4,<INDAT             I/O 5 - SHOULD FORCE STATUS BIT 10
        03EC  0000 14A5
001048  03EE  0704                             BIOT  >DNR38
001049                                  *
001050  03EF  E380 114B                         LNJ   $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001051                                  *
001052  03F1  0F99                              B     >PET                    GO TO NEXT TEST
001053                                  *
001054  03F2  1CF6                      DNR38  LDV   $R1,=-10                FORCE 10 NAKS
001055  03F3  8057                      DNR39  IO    =$R7,<OUTDAT            I/O 6-15 - SHOULD CAUSE NAKS
        03F4  0000 14A3
001056  03F6  0784                             BIOF  >DNR40
001057  03F7  E380 1180                         LNJ   $B6,<ACKERR             **  ERROR "ACK"  **
001058                                  *  GOT ACK.  SHOULD HAVE BEEN NAK'D BECAUSE BIT 10 SHOULD BE SET.
001059                                  *  REPLACE WCS MOTHER BOARD.
001060                                  *
001061  03F9  0F91                              B     >PET                    GO TO NEXT TEST
001062                                  *
001063  03FA  17F9                      DNR40  BINC  $R1,>DNR39              DO 10 TIMES
001064                                  *
001065  03FB  8054                             IO    =$R4,<INSTAT
        03FC  0000 14A6
001066  03FE  0704                             BIOT  >DNR41
001067  03FF  E380 114B                         LNJ   $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001068                                  *                                     INPUT STATUS SHOULD NEVER BE NAK'D.
```

```
001069                                  *
001070  0401  0F89                       *       B       >PET                    GO TO NEXT TEST
001071                                  *
001072  0402  D870 0020                 DNR41   LDR     $R5,=Z'0020'
001073  0404  D400 148D                         OR      $R5,<MODBIT             STATUS 5B
001074  0406  D954                              CMR     $R5,=$R4
001075  0407  0903                              BE      >PET                    IF O.K., GO TO NEXT TEST
001076                                  *
001077  0408  E380 11B5                         LNJ     $B6,<STERR              **  ERROR "STATUS"  **
001078                                  *  NAK SHOULD HAVE BEEN CAUSED BY STATUS BIT 10 (NATIVE PROM LOADING
001079                                  *  ERROR), AND ONLY BIT 10.  STATUS SHOULD BE HEX 0020, PLUS MODE BIT.
001080                                  *  STATUS READ IS IN R4.  REPLACE WCS MOTHER BOARD.
001081                                  *
001082                                  *
001083                                  *
001084                                  *
001085                                  ********************
001086                                  *                  *
001087                                  * PARITY ERROR TEST *
001088                                  *                  *
001089                                  ********************
001090                                  *
001091                                  * NOTE: PARITY CHECKED ON EVERYTHING THAT GOES OVER ON DATA BUS TO WCS
001092                                  *       EXCEPT WHEN OUTPUT CONTROL WORD.
001093                                  *
001094  040A  8980 1495                 PET     CMZ     <RAM1FL
001095  040C  0904                              BE      >PET1
001096  040D  9870 0800                         LDR     $R1,=Z'0800'            USE RAM 1 ADRS IF PRESENT
001097  040F  0F83                              B       >PET2
001098  0410  9870 0C00                 PET1    LDR     $R1,=Z'0C00'            USE RAM 2 ADRS IF NO RAM 1
001099  0412  9F00 1497                 PET2    STR     $R1,<RAMADR
001100                                  *
001101                                  * VERIFY CAN SET BIT 14 (PARITY ERROR)
001102                                  *
001103  0414  8000 1497                         IO      <RAMADR,<OUTADR
      0416  0000 14A2
001104  0418  0703                              BIOT    >PET3
001105  0419  E380 114B                         LNJ     $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001106                                  *                                       OUTPUT ADRS SHOULD NEVER BE NAK'D.
001107                                  *
001108  041B  8070 2000                 PET3    IO      =Z'2000',<OUTCON        FORCE LEFT PARITY BIT BAD
      041D  0000 14A1
001109  041F  0703                              BIOT    >PET4
001110  0420  E380 114B                         LNJ     $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001111                                  *                                       OUTPUT CONTROL WORD SHOULD NEVER BE NAK'D.
001112                                  *
001113                                  * VERIFY THAT INPUT STATUS DOES CAUSE PARITY ERROR WHEN IN PARITY
001114                                  * FORCING MODE, I.E., VERIFY THAT PARITY CHECKED ON INPUT STATUS COMMAND
001115                                  *
001116  0422  8054                      PET4    IO      =$R4,<INSTAT
      0423  0000 14A6
001117  0425  0703                              BIOT    >PET4A
001118  0426  E380 114B                         LNJ     $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001119                                  *                                       INPUT STATUS SHOULD NEVER BE NAK'D.
001120                                  *
001121  0428  D870 1002                 PET4A   LDR     $R5,=Z'1002'
001122  042A  D400 148D                         OR      $R5,<MODBIT             R5=STATUS 5B
001123  042C  C955                              CMR     $R4,=$R5
001124  042D  0903                              BE      >PET4B
001125  042E  E380 11B5                         LNJ     $B6,<STERR              **  ERROR "STATUS"  **  REPLACE WCS MB
001126                                  *  STATUS 5B HEX 1002 PLUS MODE BIT(4 OR 9).  STATUS READ IS IN R4.
001127                                  *
001128  0430  8070 0000                 PET4B   IO      =Z'0000',<OUTDAT        SHOULD SET BIT 14 (PARITY ERROR)
      0432  0000 14A3
001129                                  *                                       PARITY NORMAL 1,1 - PARITY SENT 1,0
001130                                  *                                       SHOULDN'T CAUSE NAK CAUSE INPUT STATUS
001131                                  *                                       SHOULD HAVE RESET PARITY.
001132                                  *
001133  0434  0703                              BIOT    >PET5
001134  0435  E380 114B                         LNJ     $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001135                                  *
001136  0437  8070 0000                 PET5    IO      =Z'0000',<OUTCON        RESET PARITY FORCING MODE
      0439  0000 14A1
001137  043B  8054                              IO      =$R4,<INSTAT
      043C  0000 14A6
001138  043E  0703                              BIOT    >PET6
001139  043F  E380 114B                         LNJ     $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001140                                  *                                       INPUT STATUS SHOULD NEVER BE NAK'D.
001141                                  *
001142  0441  5C02                      PET6    LDV     $R5,=2
001143  0442  D400 148D                         OR      $R5,<MODBIT
001144  0444  C955                              CMR     $R4,=$R5
001145  0445  0903                              BE      >PET7
001146  0446  E380 11B5                         LNJ     $B6,<STERR              **  ERROR "STATUS"  **  REPLACE WCS MB
001147                                  *  STATUS 5B HEX 0002 PLUS MODE BIT(4 OR 9).  STATUS READ IN R4.
001148                                  *
001149                                  *
001150                                  * VERIFY INPUT STATUS CLEARS BIT 14.
001151                                  *
001152  0448  8054                      PET7    IO      =$R4,<INSTAT
      0449  0000 14A6
001153  044B  0703                              BIOT    >PET8
001154  044C  E380 114B                         LNJ     $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001155                                  *                                       INPUT STATUS SHOULD NEVER BE NAK'D.
001156                                  *
001157  044E  D800 148D                 PET8    LDR     $R5,<MODBIT
001158  0450  C955                              CMR     $R4,=$R5
001159  0451  0903                              BE      >PET9
001160  0452  E380 11B5                         LNJ     $B6,<STERR              **  ERROR "STATUS"  **  REPLACE WCS MB
001161                                  *  INPUT STATUS SHOULD HAVE CLEARED BIT 14.  STATUS READ IN R4.
001162                                  *
001163  0454  8070 1000                 PET9    IO      =Z'1000',<OUTCON        SET PARITY FORCING MODE AGAIN
      0456  0000 14A1
001164  0458  8070 0000                         IO      =Z'0000',<OUTDAT        SHOULD SET BIT 14 AGAIN
      045A  0000 14A3
001165  045C  0703                              BIOT    >PET10
001166  045D  E380 114B                         LNJ     $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001167                                  *
001168                                  * VERIFY NAKS
001169                                  *
001170  045F  2CF6                      PET10   LDV     $R2,=-10                FORCE 10 NAKS
001171  0460  8070 0000                         IO      =Z'0000',<OUTCON        RESET PARITY FORCING MODE
      0462  0000 14A1
```

```
001172  0464  0703                        BIOT  >PET11
001173  0465  E380 114B                    LNJ   $B6,<NAKERR          **  ERROR "NAK"  **   REPLACE WCS MB
001174                              *               OUTPUT CONTROL WORD SHOULD NEVER BE NAK'D.
001175                              *
001176  0467  8070 0000              PET11 IO    =Z'0000',<OUTDAT     SHOULD CAUSE NAK
        0469  0000 14A3
001177  046B  0783                        BIOF  >PET12
001178  046C  E380 1180                    LNJ   $B6,<ACKERR          **  ERROR "ACK"  **   REPLACE WCS MB
001179                              *  GOT ACK.  SHOULD HAVE BEEN NAK'D BECAUSE ERROR FLOP(BIT 14) SHOULD
001180                              *  HAVE BEEN SET WHEN OUTPUT DATA.
001181                              *
001182  046E  27F9                  PET12 BINC  $R2,>PET11           DO 10 TIMES
001183                              *
001184  046F  8054                        IO    =$R4,<INSTAT
        0470  0000 14A6
001185  0472  0703                        BIOT  >PET13
001186  0473  E380 114B                    LNJ   $B6,<NAKERR          **  ERROR "NAK"  **   REPLACE WCS MB
001187                              *                                  INPUT STATUS SHOULD NEVER BE NAK'D.
001188                              *
001189  0475  D870 0002              PET13 LDR   $R5,=Z'0002'
001190  0477  D400 148D                    OR    $R5,<MODBIT
001191  0479  C955                        CMR   $R4,=$R5
001192  047A  0903                        BE    >PET14
001193  047B  E380 11B5                    LNJ   $B6,<STERR           **  ERROR "STATUS"  **  REPLACE WCS MB
001194                              *  STATUS SB HEX 0002 PLUS MODE BIT(4 OR 9).  STATUS READ IN R4.
001195                              *
001196                              * VERIFY INPUT DATA FORCES PARITY ERROR IN PARITY FORCING MODE,
001197                              * I.E., VERIFY PARITY CHECKED ON INPUT DATA COMMAND.
001198                              *
001199  047D  8070 3000              PET14 IO    =Z'3000',<OUTCON     FORCE PARITY FORCING MODE
        047F  0000 14A1
001200  0481  0703                        BIOT  >PET14A
001201  0482  E380 114B                    LNJ   $B6,<NAKERR          **  ERROR "NAK"  **   REPLACE WCS MB
001202                              *               OUTPUT CONTROL WORD SHOULD NEVER BE NAK'D.
001203                              *
001204  0484  8056                  PET14A IO    =$R6,<INDAT
        0485  0000 14A5
001205  0487  0703                        BIOT  >PET15
001206  0488  E380 114B                    LNJ   $B6,<NAKERR          **  ERROR "NAK"  **   REPLACE WCS MB
001207                              *
001208  048A  8070 0000              PET15 IO    =Z'0000',<OUTCON     RESET PARITY FORCING MODE
        048C  0000 14A1
001209  048E  8054                        IO    =$R4,<INSTAT
        048F  0000 14A6
001210  0491  0703                        BIOT  >PET16
001211  0492  E380 114B                    LNJ   $B6,<NAKERR          **  ERROR "NAK"  **   REPLACE WCS MB
001212                              *                                  INPUT STATUS SHOULD NEVER BE NAK'D.
001213                              *
001214  0494  5C02                  PET16 LDV   $R5,=2
001215  0495  D400 148D                    OR    $R5,<MODBIT
001216  0497  C955                        CMR   $R4,=$R5
001217  0498  0903                        BE    >PET16B
001218  0499  E380 11B5                    LNJ   $B6,<STERR           **  ERROR "STATUS"  **  REPLACE WCS MB
001219                              *  STATUS SB HEX 0002 PLUS MODE BIT(4 OR 9).  STATUS READ IN R4.
001220                              *
001221                              * VERIFY PARITY CHECKED ON INPUT I.D. COMMAND
001222                              *
001223  049B  8070 2000              PET16B IO    =Z'2000',<OUTCON     SET PARITY FORCING MODE
        049D  0000 14A1
001224  049F  8053                        IO    =$R3,<INID           INPUT I.D.  SHOULD SET BIT 14.
        04A0  0000 14A7
001225  04A2  8070 0000                    IO    =Z'0000',<OUTCON     RESET PARITY FORCING MODE
        04A4  0000 14A1
001226  04A6  8054                        IO    =$R4,<INSTAT
        04A7  0000 14A6
001227  04A9  5C02                        LDV   $R5,=2
001228  04AA  D400 148D                    OR    $R5,<MODBIT
001229  04AC  C955                        CMR   $R4,=$R5
001230  04AD  0903                        BE    >PET16C
001231  04AE  E380 11B5                    LNJ   $B6,<STERR           **  ERROR "STATUS"  **  REPLACE WCS MB
001232                              *  STATUS SB HEX 0002 PLUS MODE BIT(4 OR 9).  STATUS READ IS IN R4.
001233                              *
001234  04B0  8970 2610              PET16C CMR   $R3,=Z'2610'
001235  04B2  090E                        BE    >PET17
001236                              PET16D CALL  ZV$ER,PET16D,MSG8   **  ERROR "ID"  **   REPLACE WCS MB
        04B3  FBC0 0003
        04B5  D380 0000       X
        04B7  0F80
        04B8  04B3
        04B9  1418
001237                              *  ID SB HEX 2610.  ID READ IS IN R3.
001238                              *
001239                                    CALL  ZV$T,MSG4            PRINT "   * REPLACE MB *"
        04BA  FBC0 0003
        04BC  D380 0000       X
        04BE  0F80
        04BF  1319
001240                              *
001241                              **********
001242                              *
001243                              * TEST PARITY CHECKER BY SENDING  ALL POSSIBLE 16 BIT PATTERNS
001244                              * (65K; 0000-FFFF) TO PARITY CHECKER 4 TIMES FORCING BOTH PARITY BITS
001245                              * BAD, LEFT BAD, RIGHT BAD, AND NEITHER BAD, RESPECTIVELY.
001246                              * THIS ALSO TESTS DATA PATH TO PARITY CHECKER (BUS NOISE PROBLEMS,ETC.).
001247                              * THIS TEST TAKES ABOUT 10 SECS.
001248                              *
001249  04C0  8070 8000              PET17 IO    =Z'8000',<OUTCON     INITIALIZE WCS
        04C2  0000 14A1
001250                              *
001251                              * DEFINE REGISTER USAGE
001252                              *
001253  04C4  8757                        CL    =$R7                 R7=DATA PATTERN
001254  04C5  5C02                        LDV   $R5,=2
001255  04C6  D400 148D                    OR    $R5,<MODBIT          R5=STATUS SB FOR 1ST 3 TRIPS
001256  04C8  B800 14A3                    LDR   $R3,<OUTDAT
001257  04CA  E800 14A1                    LDR   $R6,<OUTCON
001258  04CC  9B80 14A6                    LAB   $B1,<INSTAT
001259  04CE  4C03                        LDV   $R4,=3
001260  04CF  CF00 1488                    STR   $R4,<TMP2            FLAG FOR PARITY FORCING COMBINATION
001261  04D1  9870 3000                    LDR   $R1,=Z'3000'         R1=CURRENT PARITY CONTROL WORD
001262  04D3  8051                        IO    =$R1,=$R6            FORCE BOTH PARITY BITS BAD FOR 1ST
        04D4  0056
001263                              *                                  TRIP.
```

```
001264                                          *
001265                                          *
001266  04D5  4C10                             PET18  LDV   $R4,=16             WRITE 4K PATTERNS 16 TIMES FOR
001267  04D6  CF00 1489                               STR   $R4,<TMP3           EACH PARITY COMBINATION.
001268                                          *
001269  04D8  A870 F000                         PET19  LDR   $R2,=-4096          4K 16-BIT WORDS = 1K 64-BIT WORDS
001270  04DA  8000 1497                               IO    <RAMADR,<OUTADR     OUTPUT STARTING RAM ADRS
        04DC  0000 14A2
001271  04DE  8054                                    IO    =$R4,$B1            INPUT STATUS TO RESET PARITY FLOP IN
        04DF  0001
001272                                          *                               CASE WERE IN PARITY FORCING MODE.
001273                                          *
001274  04E0  8057                             PET20  IO    =$R7,=$R3           OUTPUT DATA
        04E1  0053
001275  04E2  0703                                    BIOT  >PET20A
001276  04E3  E380 114B                               LNJ   $B6,<NAKERR         **  ERROR "NAK"  **   REPLACE WCS MB
001277                                          *
001278  04E5  8754                             PET20A CL    =$R4
001279  04E6  8054                                    IO    =$R4,=$R6           RESET PARITY FORCING MODE
        04E7  0056
001280  04E8  8054                                    IO    =$R4,$B1            INPUT STATUS
        04E9  0001
001281  04EA  C955                                    CMR   $R4,=$R5
001282  04EB  092E                                    BE    >PET22
001283                                          *
001284                                          * REPORT ERROR
001285                                          *
001286  04EC  E380 11B5                               LNJ   $B6,<STERR          **  ERROR "STATUS"  **  REPLACE WCS MB
001287                                          *  STATUS SB = R5.  STATUS READ IN R4.
001288                                          *
001289                                                CALL  ZV$TC,MSG19         PRINT "DATA ="
        04EE  FBC0 0003
        04F0  D380 0000      X
        04F2  0F80
        04F3  1453
001290  04F4  C852                                    LDR   $R4,=$R2            GET COUNT
001291  04F5  CA70 1000                               ADD   $R4,=4096
001292  04F7  8654                                    CPL   =$R4
001293  04F8  8753                                    CL    =$R3
001294  04F9  8000 1497                               IO    <RAMADR,<OUTADR     OUTPUT STARTING RAM ADRS
        04FB  0000 14A2
001295                                          *
001296  04FD  8053                             PET21  IO    =$R3,<INDAT
        04FE  0000 14A5
001297  0500  47FD                                    BINC  $R4,>PET21
001298  0501  BF00 1487                               STR   $R3,<TMP1
001299                                                CALL  ZV$THZ,TMP1         PRINT DATA READ
        0503  FBC0 0003
        0505  D380 0000      X
        0507  0F80
        0508  1487
001300                                                CALL  ZV$T,MSG7           PRINT ",  SB"
        0509  FBC0 0003
        050B  D380 0000      X
        050D  0F80
        050E  1415
001301  050F  FF00 1487                               STR   $R7,<TMP1
001302                                                CALL  ZV$THZ,TMP1         PRINT DATA SB
        0511  FBC0 0003
        0513  D380 0000      X
        0515  0F80
        0516  1487
001303  0517  B800 14A3                               LDR   $R3,<OUTDAT         RESTORE R3
001304                                          *
001305                                          * CONTINUE
001306                                          *
001307  0519  8AD7                             PET22  INC   =$R7                BUMP PATTERN
001308  051A  8051                                    IO    =$R1,=$R6           RESTORE PARITY FORCING MODE IF
        051B  0056
001309                                          *                               APPLICABLE.
001310  051C  2781 FFC3                               BINC  $R2,PET20           OUTPUT NEXT PATTERN
001311  051E  8880 1489                               DEC   <TMP3
001312  0520  8980 1489                               CMZ   <TMP3
001313  0522  0981 FFB5                               BNE   PET19               RE-INITIALIZE RAM ADRS FOR NEXT 4K
001314                                          *                               PATTERNS(DO 16 TIMES FOR EACH PARITY FORCING COMBO).
001315                                          *
001316  0524  8880 1488                               DEC   <TMP2
001317  0526  8051                                    IO    =$R1,<INID          NEED DUMMY I/O COMMAND OR ELSE IF
        0527  0000 14A7
001318                                          * NEXT I/O COMMAND IS SUPPOSED TO RESET PARITY FORCING MODE. IT WON'T.
001319                                          * INPUTTING ID HERE CAUSES PARITY ERROR IF WERE IN PARITY FORCING MODE.
001320  0529  9800 1488                               LDR   $R1,<TMP2
001321  052B  1814                                    BLZ   $R1,>MET            ALL DONE HERE.  GO TO NEXT TEST.
001322                                          *
001323  052C  1054                                    SCR   $R1,4               SHIFT PARITY FORCING BITS FROM 14,15
001324  052D  8051                                    IO    =$R1,=$R6           TO 2,3 AND OUTPUT.
        052E  0056
001325  052F  1981 FFA5                               BNEZ  $R1,PET18           OUTPUT 65K PATTERNS FOR NEXT PARITY
001326                                          *                               FORCING COMBO.
001327                                          *
001328                                          * VERIFY OUTPUT CONTROL WORD RESET PARITY FORCING MODE
001329                                          *
001330  0531  8054                                    IO    =$R4,<INSTAT        SHOULD SEE PARITY ERROR FROM INPUT ID
        0532  0000 14A6
001331  0534  5C02                                    LDV   $R5,=2
001332  0535  D400 148D                               OR    $R5,<MODBIT
001333  0537  C955                                    CMR   $R4,=$R5
001334  0538  0903                                    BE    >PET23
001335  0539  E380 11B5                               LNJ   $B6,<STERR          **  ERROR "STATUS"  **  REPLACE WCS MB
001336                                          * BITS 14(PARITY ERROR) AND MODE BIT(4 OR 9) SB SET.  STATUS READ IN R4.
001337                                          *
001338  053B  D800 148D                         PET23  LDR   $R5,<MODBIT         STATUS SHOULD ONLY SHOW MODE BIT SET
001339                                          *                               FOR LAST TRIP.
001340  053D  0F81 FF97                               B     PET18               OUTPUT 65K PATTERNS WITH NO PARITY
001341                                          *                               FORCING.
001342                                          *
001343                                          *
001344                                          *
001345                                          *
001346                                          ********************
001347                                          *                  *
001348                                          * MULTI-ERROR TEST *
001349                                          *                  *
```

```
001350                                  ********************
001351                                  *
001352                                  * VERIFY HISTORIES PRESERVED
001353                                  *
001354                                  * 1ST VERIFY CAN FORCE BIT 13(NON-EXISTANT RESOURCE) WHEN BITS 10
001355                                  * (NATIVE PROM LOADING ERROR) AND 14(PARITY ERROR) ARE SET.  ALSO VERIFY
001356                                  * THAT BITS 10 AND 14 NOT RESET WHEN OUTPUT DATA TO NON-EXISTANT
001357                                  * RESOURCE.
001358                                  *
001359  053F  8070 8000                MET     IO      =Z'8000',<OUTCON        INITIALIZE WCS
      0541  0000 14A1
001360  0543  0703                             BIOT    >MET1
001361  0544  E380 114B                        LNJ     $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001362                                  *                                       OUTPUT CONTROL WORD SHOULD NEVER BE NAK'D.
001363                                  *
001364  0546  2CFE                     MET1    LDV     $R2,=-2                 FLAG TO DETERMINE IF SHOULD INPUT
001365                                  *                                       STATUS OR INITIALIZE WCS.
001366  0547  8070 0000                MET1A   IO      =Z'0000',<OUTADR        ADRS OF NATIVE PROM
      0549  0000 14A2
001367  054B  0703                             BIOT    >MET2
001368  054C  E380 114B                        LNJ     $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001369                                  *                                       OUTPUT ADRS SHOULD NEVER BE NAK'D.
001370                                  *
001371  054E  8070 1000                MET2    IO      =Z'1000',<OUTCON        FORCE RIGHT PARITY BIT BAD
      0550  0000 14A1
001372  0552  0703                             BIOT    >MET3
001373  0553  E380 114B                        LNJ     $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001374                                  *                                       OUTPUT CONTROL WORD SHOULD NEVER BE NAK'D.
001375                                  *
001376  0555  8070 0000                MET3    IO      =Z'0000',<OUTDAT        SHOULD SET STATUS BITS 10 AND 14
      0557  0000 14A3
001377  0559  0703                             BIOT    >MET4
001378  055A  E380 114B                        LNJ     $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001379                                  *
001380  055C  8070 0000                MET4    IO      =Z'0000',<OUTCON        RESET PARITY FORCING MODE
      055E  0000 14A1
001381                                  * RESETTING PARITY FORCING MODE ALSO NECESSARY SO OUTPUTTING ADRS, ETC.,
001382                                  * FOLLOWING DOESN'T CAUSE PARITY ERROR.
001383  0560  0703                             BIOT    >MET4A
001384  0561  E380 114B                        LNJ     $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001385                                  *                                       OUTPUT CONTROL WORD SHOULD NEVER BE NAK'D.
001386                                  *
001387  0563  8070 0400                MET4A   IO      =Z'0400',<OUTADR        ADRS OF NON-EXISTANT RESOURCE
      0565  0000 14A2
001388  0567  0703                             BIOT    >MET5
001389  0568  E380 114B                        LNJ     $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001390                                  *                                       OUTPUT ADRS SHOULD NEVER BE NAK'D.
001391                                  *
001392  056A  8070 0000                MET5    IO      =Z'0000',<OUTDAT        SHOULD SET STATUS BIT 13 AND CAUSE NAK
      056C  0000 14A3
001393  056E  0783                             BIOF    >MET6
001394  056F  E380 1180                        LNJ     $B6,<ACKERR             **  ERROR "ACK"  **   REPLACE WCS MB
001395                                  *  SHOULD HAVE BEEN NAK'D CAUSE ERROR FLOPS 10 AND 14 SHOULD HAVE BEEN
001396                                  *  SET WHEN OUTPUT DATA.
001397                                  *
001398  0571  2795                     MET6    BINC    $R2,>MET9
001399                                  *
001400                                  * 2ND TIME - VERIFY INITIALIZE RESETS BITS 10,13 AND 14
001401                                  *
001402  0572  8070 8000                        IO      =Z'8000',<OUTCON        INITIALIZE WCS
      0574  0000 14A1
001403  0576  0703                             BIOT    >MET7
001404  0577  E380 114B                        LNJ     $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001405                                  *                                       OUTPUT CONTROL WORD SHOULD NEVER BE NAK'D.
001406                                  *
001407  0579  8054                     MET7    IO      =$R4,<INSTAT
      057A  0000 14A6
001408  057C  0703                             BIOT    >MET8
001409  057D  E380 114B                        LNJ     $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001410                                  *                                       INPUT STATUS SHOULD NEVER BE NAK'D.
001411                                  *
001412  057F  D800 148D                MET8    LDR     $R5,<MODBIT             STATUS SB
001413  0581  C955                             CMR     $R4,=$R5
001414  0582  0921                             BE      >MET13
001415  0583  E380 11B5                        LNJ     $B6,<STERR              **  ERROR "STATUS"  **  REPLACE WCS MB
001416                                  *  STATUS SHOULD HAVE BEEN CLEARED BY INITIALIZE COMMAND.
001417                                  *  STATUS READ IN R4.
001418                                  *
001419  0585  0F9E                             B       >MET13
001420                                  *
001421                                  * 1ST TIME - VERIFY INPUT STATUS RESETS BITS 10,13,14
001422                                  *
001423  0586  8054                     MET9    IO      =$R4,<INSTAT
      0587  0000 14A6
001424  0589  0703                             BIOT    >MET10
001425  058A  E380 114B                        LNJ     $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001426                                  *                                       INPUT STATUS SHOULD NEVER BE NAK'D.
001427                                  *
001428  058C  D870 0026                MET10   LDR     $R5,=Z'0026'
001429  058E  D400 148D                        OR      $R5,<MODBIT
001430  0590  D954                             CMR     $R5,=$R4
001431  0591  0903                             BE      >MET11
001432  0592  E380 11B5                        LNJ     $B6,<STERR              **  ERROR "STATUS"  **  REPLACE WCS MB
001433                                  *  STATUS SHOULD BE HEX 0026 PLUS MODE BIT(4 OR 9).  STATUS READ IN R4.
001434                                  *
001435  0594  8054                     MET11   IO      =$R4,<INSTAT
      0595  0000 14A6
001436  0597  0703                             BIOT    >MET12
001437  0598  E380 114B                        LNJ     $B6,<NAKERR             **  ERROR "NAK"  **   REPLACE WCS MB
001438                                  *                                       INPUT STATUS SHOULD NEVER BE NAK'D.
001439                                  *
001440  059A  D800 148D                MET12   LDR     $R5,<MODBIT             STATUS SB
001441  059C  D954                             CMR     $R5,=$R4
001442  059D  0901 FFA9                        BE      MET1A                   GO BACK FOR INITIALIZE
001443  059F  E380 11B5                        LNJ     $B6,<STERR              **  ERROR "STATUS"  **  REPLACE WCS MB
001444                                  *  INPUT STATUS SHOULD HAVE CLEARED ALL BITS EXCEPT MODE BIT(4 OR 9).
001445                                  *  STATUS READ IN R4.
001446  05A1  0F81 FFA5                        B       MET1A                   GO BACK FOR INITIALIZE
001447                                  *
001448                                  *
001449                                  * NEXT VERIFY CAN FORCE BIT 10(NATIVE PROM LOADING ERROR) WHEN BITS 13
001450                                  * (NON-EXISTANT RESOURCE) AND 14(PARITY ERROR) ARE SET.  ALSO VERIFY
001451                                  * BIT 14 NOT RESET WHEN OUTPUT DATA TO NATIVE PROM.
```

```
001452                                          *
001453  05A3  8070 0400                         MET13  IO    =Z'0400',<OUTADR      ADRS OF NON-EXISTANT RESOURCE
        05A5  0000 14A2
001454  05A7  0703                                     BIOT  >MET14
001455  05A8  E380 114B                                LNJ   $B6,<NAKERR          **  ERROR "NAK"  **   REPLACE WCS MB
001456                                          *                                OUTPUT ADRS SHOULD NEVER BE NAK'D.
001457                                          *
001458  05AA  8070 3000                         MET14  IO    =Z'3000',<OUTCON      FORCE BOTH PARITY BITS BAD
        05AC  0000 14A1
001459  05AE  0703                                     BIOT  >MET15
001460  05AF  E380 114B                                LNJ   $B6,<NAKERR          **  ERROR "NAK"  **   REPLACE WCS MB
001461                                          *                                OUTPUT CONTROL WORD SHOULD NEVER BE NAK'D.
001462                                          *
001463  05B1  8070 0000                         MET15  IO    =Z'0000',<OUTDAT      SHOULD SET STATUS BITS 13 AND 14
        05B3  0000 14A3
001464  05B5  0703                                     BIOT  >MET16
001465  05B6  E380 114B                                LNJ   $B6,<NAKERR          **  ERROR "NAK"  **   REPLACE WCS MB
001466                                          *
001467  05B8  8070 0000                         MET16  IO    =Z'0000',<OUTCON      RESET PARITY FORCING MODE
        05BA  0000 14A1
001468                                          * RESETTING PARITY FORCING MODE ALSO NECESSARY SO OUTPUTTING ADRS, ETC.,
001469                                          * FOLLOWING DOESN'T CAUSE PARITY ERROR.
001470  05BC  0703                                     BIOT  >MET17
001471  05BD  E380 114B                                LNJ   $B6,<NAKERR          **  ERROR "NAK"  **   REPLACE WCS MB
001472                                          *                                OUTPUT CONTROL WORD SHOULD NEVER BE NAK'D.
001473                                          *
001474  05BF  8070 0000                         MET17  IO    =Z'0000',<OUTADR      ADRS OF NATIVE PROM
        05C1  0000 14A2
001475  05C3  0703                                     BIOT  >MET18
001476  05C4  E380 114B                                LNJ   $B6,<NAKERR          **  ERROR "NAK"  **   REPLACE WCS MB
001477                                          *                                OUTPUT ADRS SHOULD NEVER BE NAK'D.
001478                                          *
001479  05C6  8070 0000                         MET18  IO    =Z'0000',<OUTDAT      SHOULD SET 10, RESET 13, AND NAK
        05C8  0000 14A3
001480  05CA  0783                                     BIOF  >MET19
001481  05CB  E380 1180                                LNJ   $B6,<ACKERR          **  ERROR "ACK"  **   REPLACE WCS MB
001482                                          *  GOT ACK.  SHOULD HAVE BEEN NAK'D BECAUSE ERROR FLOP(BIT 14) SHOULD
001483                                          *  HAVE BEEN SET WHEN OUTPUT DATA.
001484                                          *
001485  05CD  8054                             MET19  IO    =$R4,<INSTAT
        05CE  0000 14A6
001486  05D0  0703                                     BIOT  >MET20
001487  05D1  E380 114B                                LNJ   $B6,<NAKERR          **  ERROR "NAK"  **   REPLACE WCS MB
001488                                          *                                INPUT STATUS SHOULD NEVER BE NAK'D.
001489                                          *
001490  05D3  D870 0022                         MET20  LDR   $R5,=Z'0022'
001491  05D5  D400 148D                                OR    $R5,<MODBIT
001492  05D7  C955                                     CMR   $R4,=$R5
001493  05D8  0903                                     BE    >MET21
001494  05D9  E380 11B5                                LNJ   $B6,<STERR           **  ERROR "STATUS"  **  REPLACE WCS MB
001495                                          *  BITS 10,14 AND MODE BIT(4 OR 9) ONLY SHOULD BE SET.  STATUS READ
001496                                          *  IN R4.
001497                                          *
001498                                          *
001499                                          * VERIFY CAN SET BITS 13(NON-EXISTANT RESOURCE ERROR) AND 14(PARITY
001500                                          * ERROR) WHEN BIT 10(NATIVE PROM LOADING ERROR) IS SET.
001501                                          *
001502  05DB  8070 0000                         MET21  IO    =Z'0000',<OUTADR      OUTPUT ADRS OF NATIVE PROM
        05DD  0000 14A2
001503  05DF  0703                                     BIOT  >MET22
001504  05E0  E380 114B                                LNJ   $B6,<NAKERR          **  ERROR "NAK"  **   REPLACE WCS MB
001505                                          *                                OUTPUT ADRS SHOULD NEVER BE NAK'D.
001506                                          *
001507  05E2  8070 0000                         MET22  IO    =Z'0000',<OUTDAT      SHOULD SET STATUS BIT 10
        05E4  0000 14A3
001508  05E6  0703                                     BIOT  >MET23
001509  05E7  E380 114B                                LNJ   $B6,<NAKERR          **  ERROR "NAK"  **   REPLACE WCS MB
001510                                          *
001511  05E9  8070 0400                         MET23  IO    =Z'0400',<OUTADR      OUTPUT ADRS OF NON-EXISTANT RESOURCE
        05EB  0000 14A2
001512  05ED  0703                                     BIOT  >MET24
001513  05EE  E380 114B                                LNJ   $B6,<NAKERR          **  ERROR "NAK"  **   REPLACE WCS MB
001514                                          *                                OUTPUT ADRS SHOULD NEVER BE NAK'D.
001515                                          *
001516  05F0  8070 1000                         MET24  IO    =Z'1000',<OUTCON      FORCE RIGHT PARITY BIT BAD
        05F2  0000 14A1
001517  05F4  8070 0000                                IO    =Z'0000',<OUTDAT      SHOULD SET BITS 13,14 AND CAUSE NAK
        05F6  0000 14A3
001518  05F8  0783                                     BIOF  >MET25
001519  05F9  E380 1180                                LNJ   $B6,<ACKERR          **  ERROR "ACK"  **   REPLACE WCS MB
001520                                          *  SHOULD HAVE BEEN NAK'D CAUSE STATUS BIT 10 SHOULD HAVE BEEN SET WHEN
001521                                          *  OUTPUT DATA.
001522                                          *
001523  05FB  8070 0000                         MET25  IO    =Z'0000',<OUTCON      RESET PARITY FORCING MODE
        05FD  0000 14A1
001524  05FF  8054                                     IO    =$R4,<INSTAT
        0600  0000 14A6
001525  0602  D870 0026                                LDR   $R5,=Z'0026'
001526  0604  D400 148D                                OR    $R5,<MODBIT
001527  0606  C955                                     CMR   $R4,=$R5
001528  0607  0903                                     BE    >MET26
001529  0608  E380 11B5                                LNJ   $B6,<STERR           **  ERROR "STATUS"  **  REPLACE WCS MB
001530                                          *  STATUS SB HEX 0026 PLUS MODE BIT(4 OR 9).  STATUS READ IS IN R4.
001531                                          *
001532                                          *
001533                                          * VERIFY PARITY CHECKED WHEN OUTPUT ADRS
001534                                          *
001535  060A  8070 3000                         MET26  IO    =Z'3000',<OUTCON      SET PARITY FORCING MODE
        060C  0000 14A1
001536  060E  8000 1497                                IO    <RAMADR,<OUTADR       SHOULD CAUSE PARITY ERROR
        0610  0000 14A2
001537  0612  0703                                     BIOT  >MET27
001538  0613  E380 114B                                LNJ   $B6,<NAKERR          **  ERROR "NAK"  **   REPLACE WCS MB
001539                                          *                                OUTPUT ADRS SHOULD NEVER BE NAK'D.
001540                                          *
001541  0615  8070 0000                         MET27  IO    =Z'0000',<OUTCON      RESET PARITY FORCING MODE
        0617  0000 14A1
001542  0619  8054                                     IO    =$R4,<INSTAT
        061A  0000 14A6
001543  061C  5C02                                     LDV   $R5,=2
001544  061D  D400 148D                                OR    $R5,<MODBIT
001545  061F  C955                                     CMR   $R4,=$R5
001546  0620  0903                                     BE    >APT                 IF O.K., GO TO NEXT TEST
```

```
001547                          *
001548  0621  E380 11B5         *       LNJ   $B6,<STERR          ** ERROR "STATUS" **  REPLACE WCS MB
001549                          *  STATUS SB HEX 0002 PLUS MODE BIT(4 OR 9).  STATUS READ IS IN R4.
001550                          *
001551                          *
001552                          *
001553                          *
001554                          ********************
001555                          *                  *
001556                          * ADDRESS PATH TEST *
001557                          *                  *
001558                          ********************
001559                          *
001560                          * THIS TEST NECESSARY ONLY IF BOTH RAMS PRESENT.
001561                          * THIS TEST ASSUMES PARITY CHECKER AND DATA PATH (DATA BUS) TO PARITY
001562                          * CHECKER HAVE BEEN TESTED.
001563                          *
001564  0623  8980 1495         APT     CMZ   <RAM1FL
001565  0625  0900 06E5                 BE    <DPT                SKIP THIS TEST.  RAM 1 NOT PRESENT.
001566  0627  8980 1496                 CMZ   <RAM2FL
001567  0629  0900 06E5                 BE    <DPT                SKIP THIS TEST.  RAM 2 NOT PRESENT.
001568                          *
001569  062B  8700 1498                 CL    <APTEF1             CLEAR ERROR FLAG 1
001570  062D  8700 1499                 CL    <APTEF2             CLEAR ERROR FLAG 2
001571                          *
001572                          * CLEAR 1ST LOC., ZONE 1 OF EACH DB
001573                          *
001574  062F  8757                     CL    =$R7
001575  0630  8070 0800                IO    =Z'0800',<OUTADR    ADRS OF 1ST RAM
      0632  0000 14A2
001576  0634  8057                     IO    =$R7,<OUTDAT
      0635  0000 14A3
001577  0637  8070 0C00                IO    =Z'0C00',<OUTADR    ADRS OF 2ND RAM
      0639  0000 14A2
001578  063B  8057                     IO    =$R7,<OUTDAT
      063C  0000 14A3
001579                          *
001580                          * WRITE HEX AAAA TO 1ST LOC., ZONE 1 OF DB 1 (MAY BE DB 2)
001581                          *
001582  063E  E870 AAAA                 LDR   $R6,=Z'AAAA'
001583  0640  8070 0800                IO    =Z'0800',<OUTADR    ADRS OF DB 1
      0642  0000 14A2
001584  0644  8056                     IO    =$R6,<OUTDAT        WRITE AAAA
      0645  0000 14A3
001585                          *
001586                          * READ 1ST LOC., ZONE 1 OF DB 1 (MAY BE 2)
001587                          *
001588  0647  8070 0800                IO    =Z'0800',<OUTADR    ADRS OF DB 1
      0649  0000 14A2
001589  064B  8054                     IO    =$R4,<INDAT         READ
      064C  0000 14A5
001590  064E  C956                     CMR   $R4,=$R6            =AAAA?
001591  064F  0905                     BE    >APT1               BRANCH IF YES
001592                          *
001593                          * SET ERROR FLAG 1.  MAY BE MB ERROR BECAUSE EITHER:  1. CAN'T WRITE
001594                          * EITHER BOARD (*);  2. MB SHORT OR OPEN (*);  OR 3. BAD DATA-IN REG.,
001595                          * DATA-OUT REG., OR DATA SELECTOR(*).  MAY BE DB ERROR BECAUSE EITHER:
001596                          * 1. SHORT ON DB 1 OR 2 (*);  2. DB 1 OPEN;  OR 3. BAD LOC ON DB 1.
001597                          * (*) DENOTES WILL OCCUR AGAIN BELOW.
001598                          *
001599  0650  1C01                     LDV   $R1,=1              SET ERROR FLAG 1
001600  0651  9F00 1498                STR   $R1,<APTEF1
001601  0653  0F8D                     B     >APT2
001602                          *
001603                          * READ 1ST LOC., ZONE 1 OF DB 2 (MAY BE DB 1)
001604                          *
001605  0654  8070 0C00         APT1   IO    =Z'0C00',<OUTADR    ADRS OF DB 2
      0656  0000 14A2
001606  0658  8054                     IO    =$R4,<INDAT         READ
      0659  0000 14A5
001607  065B  C956                     CMR   $R4,=$R6            =AAAA?
001608  065C  0984                     BNE   >APT2               BRANCH IF NO
001609                          *
001610                          * SET ERROR FLAG 2.  EITHER DB 2 ERROR;  WRITING INTO BOARD 1 CAUSES
001611                          * WRITING INTO BOARD 2; OR MB ERROR:  ALWAYS SELECT SAME DB (IF MB, WILL
001612                          * OCCUR AGAIN BELOW).
001613                          *
001614  065D  1C01                     LDV   $R1,=1              SET ERROR FLAG 2
001615  065E  9F00 1499                STR   $R1,<APTEF2
001616                          *
001617                          * CLEAR 1ST LOC., ZONE 1 OF EACH DB
001618                          *
001619  0660  8070 0800         APT2   IO    =Z'0800',<OUTADR    ADRS OF 1ST RAM
      0662  0000 14A2
001620  0664  8057                     IO    =$R7,<OUTDAT
      0665  0000 14A3
001621  0667  8070 0C00                IO    =Z'0C00',<OUTADR    ADRS OF 2ND RAM
      0669  0000 14A2
001622  066B  8057                     IO    =$R7,<OUTDAT
      066C  0000 14A3
001623                          *
001624                          * WRITE HEX AAAA TO 1ST LOC., ZONE 1 OF DB 2 (MAY BE DB 1)
001625                          *
001626  066E  8070 0C00                IO    =Z'0C00',<OUTADR    ADRS OF DB 2
      0670  0000 14A2
001627  0672  8056                     IO    =$R6,<OUTDAT        WRITE AAAA
      0673  0000 14A3
001628                          *
001629                          * READ 1ST LOC., ZONE 1 OF DB 2 (MAY BE DB 1)
001630                          *
001631  0675  8070 0C00                IO    =Z'0C00',<OUTADR    ADRS OF DB 2
      0677  0000 14A2
001632  0679  8054                     IO    =$R4,<INDAT         READ
      067A  0000 14A5
001633  067C  C956                     CMR   $R4,=$R6            =AAAA?
001634  067D  0920                     BE    >APT5               BRANCH IF YES
001635                          *
001636  067E  8754                     CL    =$R4                ERROR
001637  067F  8054                     IO    =$R4,<INSTAT
      0680  0000 14A6
001638  0682  C900 148D                CMR   $R4,<MODBIT
001639  0684  0981 005C                BNE   APT15
001640  0686  8980 1498                CMZ   <APTEF1             ERROR FLAG 1 SET?
```

```
001641  0688  090C                        BE    >APT4               BRANCH IF NO
001642                                    *
001643                                    APT3  CALL  ZV$ER,APT3,MAPT1    **  ERROR "APT1"  **  MAY BE WCS MB
        0689  FBC0 0003
        068B  D380 0000       X
        068D  0F80
        068E  0689
        068F  13F2
001644                                    *  EITHER MB ERROR (SEE COMMENT ABOVE REGARDING SETTING OF ERROR FLAG 1)
001645                                    *  OR SHORT ON EITHER DB.
001646                                    *
001647  0690  E380 1205                        LNJ   $B6,<PRMAY          PRINT "   * MAY BE WCS MB *"
001648  0692  0F81 004B                        B     APT14
001649                                    *
001650                                    APT4  CALL  ZV$ER,APT4,MAPT2    **  ERROR "APT2"  **   REPLACE DB 2
        0694  FBC0 0003
        0696  D380 0000       X
        0698  0F80
        0699  0694
        069A  13F5
001651                                    *                               EITHER DB 2 OPEN OR BAD LOC ON DB 2
001652                                    *
001653  069B  0F81 0040                        B     APT13
001654                                    *
001655                                    * READ 1ST LOC., ZONE 1 OF DB 1 (MAY BE DB 2)
001656                                    *
001657  069D  8070 0800                   APT5  IO    =Z'0800',<OUTADR      ADRS OF DB 1
        069F  0000 14A2
001658  06A1  8053                             IO    =$R3,<INDAT           READ
        06A2  0000 14A5
001659  06A4  8754                             CL    =$R4
001660  06A5  8054                             IO    =$R4,<INSTAT
        06A6  0000 14A6
001661  06A8  C900 148D                        CMR   $R4,<MODBIT
001662  06AA  0981 0036                        BNE   APT15
001663  06AC  B956                             CMR   $R3,=$R6              =AAAA?
001664  06AD  0996                             BNE   >APT8               BRANCH IF NO
001665                                    *
001666  06AE  8980 1499                        CMZ   <APTEF2             ERROR.  IS ERROR FLAG 2 SET?
001667  06B0  090B                             BE    >APT7               BRANCH IF NO
001668
001669                                    APT6  CALL  ZV$ER,APT6,MAPT3    **  ERROR "APT3"  **   REPLACE WCS MB
        06B1  FBC0 0003
        06B3  D380 0000       X
        06B5  0F80
        06B6  06B1
        06B7  13F8
001670                                    *                               ALWAYS SELECT SAME DB.
001671                                    *
001672  06B8  E380 11F0                        LNJ   $B6,<PRRMB          PRINT "   * REPLACE WCS MB *"
001673  06BA  0FA4                             B     >APT14
001674                                    *
001675                                    APT7  CALL  ZV$ER,APT7,MAPT4    **  ERROR "APT4"  **   REPLACE DB 1
        06BB  FBC0 0003
        06BD  D380 0000       X
        06BF  0F80
        06C0  06BB
        06C1  13FB
001676                                    *  WRITING INTO DB 2 CAUSES WRITING INTO DB 1.
001677                                    *
001678  06C2  0F97                             B     >APT12
001679
001680  06C3  8980 1498                   APT8  CMZ   <APTEF1             ERROR FLAG 1 SET?
001681  06C5  0909                             BE    >APT10              BRANCH IF NO
001682                                    *
001683                                    APT9  CALL  ZV$ER,APT9,MAPT5    **  ERROR "APT5"  **   REPLACE DB 1
        06C6  FBC0 0003
        06C8  D380 0000       X
        06CA  0F80
        06CB  06C6
        06CC  13FE
001684                                    *                               EITHER DB 1 OPEN OR BAD LOC ON DB 1
001685  06CD  0F8C                             B     >APT12
001686                                    *
001687  06CE  8980 1499                   APT10 CMZ   <APTEF2             ERROR FLAG 2 SET?
001688  06D0  0915                             BE    >DPT                BRANCH IF NO TO NEXT TEST
001689                                    *
001690                                    APT11 CALL  ZV$ER,APT11,MAPT6   **  ERROR "APT6"  **   REPLACE DB 2
        06D1  FBC0 0003
        06D3  D380 0000       X
        06D5  0F80
        06D6  06D1
        06D7  1401
001691                                    *  WRITING INTO DB 1 CAUSES WRITING INTO DB 2
001692                                    *
001693  06D8  0F84                             B     >APT13
001694                                    *
001695  06D9  E380 11F7                   APT12 LNJ   $B6,<PRRDB1           PRINT "   * REPLACE DB 1 *"
001696  06DB  0F83                             B     >APT14
001697                                    *
001698  06DC  E380 11FE                   APT13 LNJ   $B6,<PRRDB2           PRINT "   * REPLACE DB 2 *"
001699                                    *
001700  06DE  E380 120C                   APT14 LNJ   $B6,<PRSTS            PRINT STATUS
001701  06E0  0F85                             B     >DPT                GO TO NEXT TEST
001702                                    *
001703  06E1  D800 148D                   APT15 LDR   $R5,<MODBIT           STATUS SB
001704  06E3  E380 11B5                        LNJ   $B6,<STERR            **  ERROR "STATUS"  ** REPLACE WCS MB
001705                                    *
001706                                    *
001707                                    *
001708                                    *
001709                                    ******************
001710                                    *                *
001711                                    * DATA PATH TEST *
001712                                    *                *
001713                                    ******************
001714                                    *
001715                                    *
001716                                    * PURPOSE OF THIS TEST TO DETECT FOLLOWING FAULTS AND, IF POSSIBLE,
001717                                    * ISOLATE TO ORU, BY WALKING 1'S AND 0'S THRU 1ST 64-BIT WORD OF EACH
001718                                    * RAM PRESENT.  CONCERNED HERE WITH PROBLEMS EXTERNAL TO MEMORY CHIPS.
001719                                    *
001720                                    *  MOTHER BOARD FAULTS:
```

```
001721                                  *       - DATA LINE SHORT/OPEN
001722                                  *       - BAD DATA-IN REG., DATA-OUT REG., OR DATA SELECTOR
001723                                  *         (E.G. BIT STUCK HIGH/LOW)
001724                                  *       - MALFUNCTIONING DATA SELECTOR
001725                                  *
001726                                  *  DAUGHTER BOARD FAULTS THAT CAN'T ISOLATE TO ORU:
001727                                  *       - DATA LINE SHORT
001728                                  *
001729                                  *  DAUGHTER BOARD FAULTS THAT CAN ISOLATE TO ORU IF HAVE 2 RAMS:
001730                                  *       - DATA LINE OPEN
001731                                  *       - BAD MEMORY CHIP
001732                                  *
001733                                  *
001734                                  * DEFINE REGISTER USAGE
001735                                  *
001736  06E5  9800 14A2                 DPT     LDR   $R1,<OUTADR
001737  06E7  A800 14A3                         LDR   $R2,<OUTDAT
001738  06E9  B800 14A5                         LDR   $R3,<INDAT
001739                                  *               $R4 USED FOR WALKING 1'S AND 0'S
001740                                  *               $R5 USED FOR SHIFT COUNT (=16)
001741                                  *               $R6 USED FOR DATA INPUT AND TEMP
001742                                  *               $R7 = 1ST LOC OF RAM 1 OR RAM 2
001743                                  *
001744                                  *
001745  06EB  8700 149A                         CL    <DPTEF1                 CLEAR RAM 1 ERROR FLAG
001746  06ED  8980 1495                         CMZ   <RAM1FL                 CHECK IF RAM 1 PRESENT
001747  06EF  0901 0062                         BE    DPT7                    BRANCH IF NO
001748                                  *
001749                                  * WALK 1'S THRU 1ST 64-BIT LOCATION OF RAM 1
001750                                  *
001751  06F1  F870 0800                         LDR   $R7,=Z'0800'            1ST LOC OF RAM 1
001752                                  *
001753  06F3  6C01                      DPT1    LDV   $R6,=1
001754  06F4  EF00 1487                         STR   $R6,<TMP1               FLAG=1 WHILE WALKING 1'S, =0 WHILE
001755                                  *                                     WALKING 0'S.
001756  06F6  4C01                              LDV   $R4,=1                  WALK 1'S FIRST.
001757                                  *
001758                                  * ZONE 4 (COMES BACK HERE TO WALK 0'S THRU 1ST LOC OF EACH RAM PRESENT)
001759                                  *
001760  06F7  5CF0                      DPT2    LDV   $R5,=-16
001761  06F8  8057                      DPT3    IO    =$R7,=$R1               OUTPUT ADRS
        06F9  0051
001762  06FA  8056                              IO    =$R6,=$R3               INPUT DATA (DUMMY FOR ZONE BUMP)
        06FB  0053
001763  06FC  8056                              IO    =$R6,=$R3               INPUT DATA (DUMMY FOR ZONE BUMP)
        06FD  0053
001764  06FE  8056                              IO    =$R6,=$R3               INPUT DATA (DUMMY FOR ZONE BUMP)
        06FF  0053
001765  0700  8054                              IO    =$R4,=$R2               OUTPUT DATA TO ZONE 4
        0701  0052
001766  0702  8057                              IO    =$R7,=$R1               OUTPUT ADRS
        0703  0051
001767  0704  8056                              IO    =$R6,=$R3               INPUT DATA (DUMMY FOR ZONE BUMP)
        0705  0053
001768  0706  8056                              IO    =$R6,=$R3               INPUT DATA (DUMMY FOR ZONE BUMP)
        0707  0053
001769  0708  8056                              IO    =$R6,=$R3               INPUT DATA (DUMMY FOR ZONE BUMP)
        0709  0053
001770  070A  8056                              IO    =$R6,=$R3               INPUT DATA FROM ZONE 4
        070B  0053
001771  070C  C956                              CMR   $R4,=$R6
001772  070D  0980 077B                         BNE   <DPTERR                 ERROR
001773  070F  4011                              SCL   $R4,1                   ROTATE 1 (OR 0) 1 POSITION LEFT
001774  0710  57E8                              BINC  $R5,>DPT3               DO 16 TIMES
001775                                  *
001776                                  * ZONE 3
001777                                  *
001778  0711  5CF0                              LDV   $R5,=-16
001779  0712  8057                      DPT4    IO    =$R7,=$R1               OUTPUT ADRS
        0713  0051
001780  0714  8056                              IO    =$R6,=$R3               INPUT DATA (DUMMY FOR ZONE BUMP)
        0715  0053
001781  0716  8056                              IO    =$R6,=$R3               INPUT DATA (DUMMY FOR ZONE BUMP)
        0717  0053
001782  0718  8054                              IO    =$R4,=$R2               OUTPUT DATA TO ZONE 3
        0719  0052
001783  071A  8057                              IO    =$R7,=$R1               OUTPUT ADRS
        071B  0051
001784  071C  8056                              IO    =$R6,=$R3               INPUT DATA (DUMMY FOR ZONE BUMP)
        071D  0053
001785  071E  8056                              IO    =$R6,=$R3               INPUT DATA (DUMMY FOR ZONE BUMP)
        071F  0053
001786  0720  8056                              IO    =$R6,=$R3               INPUT DATA FROM ZONE 3
        0721  0053
001787  0722  C956                              CMR   $R4,=$R6
001788  0723  0980 077B                         BNE   <DPTERR                 ERROR
001789  0725  4011                              SCL   $R4,1                   ROTATE 1 (OR 0) 1 POSITION LEFT
001790  0726  57EC                              BINC  $R5,>DPT4               DO 16 TIMES
001791                                  *
001792                                  * ZONE 2
001793                                  *
001794  0727  5CF0                              LDV   $R5,=-16
001795  0728  8057                      DPT5    IO    =$R7,=$R1               OUTPUT ADRS
        0729  0051
001796  072A  8056                              IO    =$R6,=$R3               INPUT DATA (DUMMY FOR ZONE BUMP)
        072B  0053
001797  072C  8054                              IO    =$R4,=$R2               OUTPUT DATA TO ZONE 2
        072D  0052
001798  072E  8057                              IO    =$R7,=$R1               OUTPUT ADRS
        072F  0051
001799  0730  8056                              IO    =$R6,=$R3               INPUT DATA (DUMMY FOR ZONE BUMP)
        0731  0053
001800  0732  8056                              IO    =$R6,=$R3               INPUT DATA FROM ZONE 2
        0733  0053
001801  0734  C956                              CMR   $R4,=$R6
001802  0735  0980 077B                         BNE   <DPTERR                 ERROR
001803  0737  4011                              SCL   $R4,1                   ROTATE 1 (OR 0) 1 POSITION LEFT
001804  0738  57F0                              BINC  $R5,>DPT5               DO 16 TIMES
001805                                  *
001806                                  * ZONE 1
001807                                  *
001808  0739  5CF0                              LDV   $R5,=-16
001809  073A  8057                      DPT6    IO    =$R7,=$R1               OUTPUT ADRS
```

```
                073B  0051
001810          073C  8054                    IO    =$R4,=$R2           OUTPUT DATA TO ZONE 1
                073D  0052
001811          073E  8057                    IO    =$R7,=$R1           OUTPUT ADRS
                073F  0051
001812          0740  8056                    IO    =$R6,=$R3           INPUT DATA FROM ZONE 1
                0741  0053
001813          0742  C956                    CMR   $R4,=$R6
001814          0743  0981 0037               BNE   DPTERR              ERROR
001815          0745  4011                    SCL   $R4,1               ROTATE 1 (OR 0) 1 POSITION LEFT
001816          0746  57F4                    BINC  $R5,>DPT6           DO 16 TIMES
001817                                  *
001818                                  * WALK 0'S THRU 1ST 64-BIT LOCATION OF THIS RAM IF HAVEN'T ALREADY
001819                                  *
001820          0747  C870 FFFE               LDR   $R4,=Z'FFFE'        PATTERN FOR WALKING 0'S
001821          0749  8880 1487               DEC   <TMP1
001822          074B  8980 1487               CMZ   <TMP1
001823          074D  0901 FFA9               BE    DPT2                IF TMP1=0, WALK 0'S
001824                                  *
001825                                  *
001826          074F  F970 0800               CMR   $R7,=Z'0800'        WERE WE TESTING RAM 1?
001827          0751  0988                    BNE   >DPT9               BRANCH IF NO
001828                                  *
001829          0752  8980 1496       DPT7    CMZ   <RAM2FL             CHECK IF RAM 2 PRESENT
001830          0754  0908                    BE    >DPT10              BRANCH IF NOT
001831                                  *
001832                                  * WALK 1'S THRU 1ST 64-BIT LOCATION OF RAM 2
001833                                  *
001834          0755  F870 0C00       DPT8    LDR   $R7,=Z'0C00'        1ST LOC OF RAM 2
001835          0757  0F81 FF9B               B     DPT1
001836                                  *
001837                                  * FINISH UP (AND CHECK FOR DB 1 ERROR)
001838                                  *
001839          0759  8980 149A       DPT9    CMZ   <DPTEF1             ERROR IN 1ST RAM?
001840          075B  098E                    BNE   >DPT12              BRANCH IF YES
001841          075C  8054              DPT10 IO    =$R4,<INSTAT        INPUT STATUS
                075D  0000 14A6
001842          075F  C900 148D               CMR   $R4,<MODBIT
001843          0761  0900 07CA               BE    <MAT                IF O.K., GO TO NEXT TEST.
001844          0763  D800 148D       DPT11   LDR   $R5,<MODBIT         STATUS SB
001845          0765  E380 11B5               LNJ   $B6,<STERR          **  ERROR "STATUS"  ** REPLACE WCS MB
001846          0767  0F81 0062               B     MAT                 GO TO NEXT TEST
001847                                  *
001848                                  * BOTH RAMS PRESENT
001849                                  *
001850          0769  8754              DPT12 CL    =$R4
001851          076A  8054                    IO    =$R4,<INSTAT
                076B  0000 14A6
001852          076D  C900 148D               CMR   $R4,<MODBIT
001853          076F  09F4                    BNE   >DPT11
001854                                  DPT13 CALL  ZV$ER,DPT13,MDPT1   **  ERROR "DPT1"  **   REPLACE DB 1
                0770  FBC0 0003
                0772  D380 0000       X
                0774  0F80
                0775  0770
                0776  1404
001855                                  *             CAUSE: EITHER DATA LINE OPEN OR BAD MEMORY CHIP.
001856                                  *
001857          0777  E380 11F7               LNJ   $B6,<PRRDB1         PRINT "   * REPLACE DB 1 *"
001858          0779  0F80 07C8               B     <DPTER5             PRINT STATUS
001859                                  *
001860                                  * ERROR ANALYSIS
001861                                  *
001862          077B  CF00 1488       DPTERR STR  $R4,<TMP2
001863          077D  8754                    CL    =$R4
001864          077E  8054                    IO    =$R4,<INSTAT
                077F  0000 14A6
001865          0781  C900 148D               CMR   $R4,<MODBIT
001866          0783  09E0                    BNE   >DPT11
001867          0784  C800 1488               LDR   $R4,<TMP2
001868                                  *
001869          0786  F970 0800               CMR   $R7,=Z'0800'        WERE WE TESTING RAM 1?
001870          0788  091C                    BE    >DPTER3             BRANCH IF YES
001871          0789  8980 1495               CMZ   <RAM1FL             NO - IS RAM 1 PRESENT?
001872          078B  0921                    BE    >DPTER4             BRANCH IF NOT
001873          078C  8980 149A               CMZ   <DPTEF1             YES - ERROR FROM 1ST RAM TEST?
001874          078E  098C                    BNE   >DPTER2             BRANCH IF YES
001875                                  *
001876                                  * BOTH RAMS PRESENT
001877                                  *
001878                                  DPTER1 CALL ZV$ER,DPTER1,MDPT2  **  ERROR "DPT2"  **   REPLACE DB 2
                078F  FBC0 0003
                0791  D380 0000       X
                0793  0F80
                0794  078F
                0795  1407
001879                                  *             CAUSE: EITHER DATA LINE OPEN OR BAD MEMORY CHIP.
001880                                  *
001881          0796  E380 11FE               LNJ   $B6,<PRRDB2         PRINT "   * REPLACE DB 2 *"
001882          0798  0F81 002F               B     DPTER5              PRINT STATUS
001883                                  *
001884                                  * BOTH RAMS PRESENT
001885                                  *
001886                                  DPTER2 CALL ZV$ER,DPTER2,MDPT3  **  ERROR "DPT3"  **   MAY BE WCS MB
                079A  FBC0 0003
                079C  D380 0000       X
                079E  0F80
                079F  079A
                07A0  140A
001887                                  *  SEE COMMENTS AT BEGINNING OF THIS TEST.  IF NOT MOTHER BOARD FAULT,
001888                                  *  THEN IT'S A DATA LINE SHORT ON ONE OF THE DAUGHTER BOARDS.
001889                                  *
001890          07A1  E380 1205               LNJ   $B6,<PRMAY          PRINT "   * MAY BE WCS MB *"
001891          07A3  0FA5                    B     >DPTER5             PRINT STATUS
001892                                  *
001893          07A4  8980 1496       DPTER3  CMZ   <RAM2FL             IS RAM 2 PRESENT?
001894          07A6  0906                    BE    >DPTER4             BRANCH IF NOT
001895          07A7  6C01                    LDV   $R6,=1              YES
001896          07A8  EF00 149A               STR   $R6,<DPTEF1         SET FLAG
001897          07AA  0F80 0755               B     <DPT8               GO TEST 2ND RAM
001898                                  *
001899                                  * JUST 1 RAM PRESENT
001900                                  *
```

```
001901                          DPTER4 CALL  ZV$ER,DPTER4,MDPT4   **  ERROR "DPT4"  **   EITHER MB OR DB
        07AC  FBC0 0003
        07AE  D380 0000       X
        07B0  0F80
        07B1  07AC
        07B2  140D
001902                          *  SEE COMMENTS AT BEGINNING OF THIS TEST.  IF NOT MOTHER BOARD FAULT,
001903                          *  THEN IT'S A FAULT ON THE DAUGHTER BOARD CAUSED EITHER BY DATA LINE
001904                          *  SHORT/OPEN OR BAD MEMORY CHIP.
001905                          *   NOTE: IF ERROR RESTRICTED TO ONE BIT OF ONE ZONE IN MAT ERRORS
001906                          *         FOLLOWING, MOST LIKELY BAD MEMORY CHIP ON DB.
001907                          *
001908                                CALL  ZV$T,MSG4D            PRINT "   * EITHER WCS MB OR DB *"
        07B3  FBC0 0003
        07B5  D380 0000       X
        07B7  0F80
        07B8  1341
001909  07B9  E380 120C               LNJ   $B6,<PRSTS            PRINT STATUS
001910                                CALL  ZV$TC,MSG4E           PRINT NOTE ABOVE.
        07BB  FBC0 0003
        07BD  D380 0000       X
        07BF  0F80
        07C0  134F
001911                                CALL  ZV$TC,MSG4F           PRINT REST OF NOTE ABOVE.
        07C1  FBC0 0003
        07C3  D380 0000       X
        07C5  0F80
        07C6  1369
001912  07C7  0F83                    B     >MAT                  GO TO NEXT TEST
001913                          *
001914                          * PRINT STATUS
001915                          *
001916  07C8  E380 120C         DPTER5 LNJ   $B6,<PRSTS            PRINT STATUS
001917                          *
001918                          *
001919                          *
001920                          *
001921                          **********************
001922                          *                    *
001923                          * MEMORY ARRAY TEST  *
001924                          *                    *
001925                          **********************
001926                          *
001927                          * THIS TEST("MAT") TAKES ABOUT 3 SECS.
001928                          * TEST PATTERNS HERE TO PICK UP ANY INTERNAL CHIP PROBLEMS.
001929                          *
001930                          * ESTABLISH "TMP3"
001931                          *
001932  07CA  1C02             MAT    LDV   $R1,=2                DO TEST TWICE.  SWITCH PATTERNS
001933  07CB  9F00 1489               STR   $R1,<TMP3             2ND TIME.
001934                          *
001935                          *
001936                          * CLEAR MEMORY
001937                          * (1ST ESTABLISH IF 1K OR 2K PRESENT - ESTABLISH "TMP1")
001938  07CD  9800 1495               LDR   $R1,<RAM1FL
001939  07CF  9A00 1496               ADD   $R1,<RAM2FL
001940  07D1  1B08                    BEVN  $R1,>MAT2
001941                          *
001942  07D2  9870 0400               LDR   $R1,=1024             1K PRESENT
001943  07D4  9F00 1487               STR   $R1,<TMP1
001944  07D6  9870 F000               LDR   $R1,=-4096
001945  07D8  0F87                    B     >MAT3
001946                          *
001947  07D9  9870 0800         MAT2  LDR   $R1,=2048             2K PRESENT
001948  07DB  9F00 1487               STR   $R1,<TMP1
001949  07DD  9870 E000               LDR   $R1,=-8192
001950                          *
001951  07DF  8000 1497         MAT3  IO    <RAMADR,<OUTADR       OUTPUT ADRS OF 1ST LOC.
        07E1  0000 14A2
001952  07E3  8752                    CL    =$R2
001953  07E4  B800 14A3               LDR   $R3,<OUTDAT
001954  07E6  8052             MAT4  IO    =$R2,=$R3             OUTPUT DATA (=0)
        07E7  0053
001955  07E8  17FE                    BINC  $R1,>MAT4             DO 4K OR 8K TIMES
001956                          *
001957  07E9  8054                    IO    =$R4,<INSTAT
        07EA  0000 14A6
001958  07EC  D800 148D               LDR   $R5,<MODBIT
001959  07EE  C955                    CMR   $R4,=$R5
001960  07EF  0903                    BE    >MAT4B
001961  07F0  E380 11B5               LNJ   $B6,<STERR            **  ERROR "STATUS"  ** REPLACE WCS MB
001962                          *
001963                          *
001964                          * DEFINE REGISTER USAGE
001965                          *
001966  07F2  9800 14A2         MAT4B LDR   $R1,<OUTADR
001967  07F4  A800 14A3               LDR   $R2,<OUTDAT
001968  07F6  B800 14A5               LDR   $R3,<INDAT
001969  07F8  C870 5555               LDR   $R4,=Z'5555'          2ND TIME = AAAA
001970  07FA  D870 AAAA               LDR   $R5,=Z'AAAA'          2ND TIME = 5555
001971                          *            $R6 USED FOR INPUT AND TEMPORARY STORE
001972                          *            $R7 DEFINED NEXT
001973  07FC  9B80 1488               LAB   $B1,<TMP2             ADRS OF COUNTER (1K OR 2K)
001974                          *
001975                          *****
001976                          *
001977                          * COMES HERE FOR 2ND TIME THRU "MAT"
001978                          *
001979                          *****
001980                          *
001981                          * SETUP FOR NEXT TEST
001982                          *
001983  07FE  F800 1497         MAT4A LDR   $R7,<RAMADR           ADRS OF 1ST RAM LOC
001984  0800  E800 1487               LDR   $R6,<TMP1
001985  0802  EF00 1488               STR   $R6,<TMP2             INITIALIZE COUNTER (1K OR 2K)
001986                          *
001987                          *****
001988                          *
001989                          * COMES HERE 1K OR 2K TIMES (DEPENDING ON NO. RAMS).
001990                          * WRITE AND VERIFY EVERY LOCATION, STARTING AT 1ST RAM LOC..
001991                          *
001992                          *****
001993                          *
```

```
001994                                  * ZONE 1
001995                                  *
001996  0804  8057                      MAT5    IO      =$R7,=$R1               OUTPUT ADRS
        0805  0051
001997  0806  8054                              IO      =$R4,=$R2               OUTPUT DATA TO ZONE 1
        0807  0052
001998  0808  8057                              IO      =$R7,=$R1               OUTPUT ADRS
        0809  0051
001999  080A  8056                              IO      =$R6,=$R3               INPUT DATA FROM ZONE 1
        080B  0053
002000  080C  C956                              CMR     $R4,=$R6
002001  080D  0905                              BE      >MAT6
002002                                  *
002003  080E  2C01                              LDV     $R2,=1                  PASS ZONE NO. (R2 RESTORED)
002004  080F  B854                              LDR     $R3,=$R4                PASS DATA S.B. (R3 RESTORED)
002005  0810  E380 0970                        LNJ     $B6,<MATERR             **  ERROR "MAT"  **    REPLACE DB IN R3
002006                                  *
002007                                  * ZONE 2
002008                                  *
002009  0812  8054                      MAT6    IO      =$R4,=$R2               OUTPUT DATA TO ZONE 2
        0813  0052
002010  0814  8057                              IO      =$R7,=$R1               OUTPUT ADRS
        0815  0051
002011  0816  8056                              IO      =$R6,=$R3               INPUT DATA (DUMMY FOR ZONE BUMP)
        0817  0053
002012  0818  8056                              IO      =$R6,=$R3               INPUT DATA FROM ZONE 2
        0819  0053
002013  081A  C956                              CMR     $R4,=$R6
002014  081B  0905                              BE      >MAT7
002015                                  *
002016  081C  2C02                              LDV     $R2,=2                  PASS ZONE NO. (R2 RESTORED)
002017  081D  B854                              LDR     $R3,=$R4                PASS DATA S.B. (R3 RESTORED)
002018  081E  E380 0970                        LNJ     $B6,<MATERR             **  ERROR "MAT"  **    REPLACE DB IN R3
002019                                  *
002020                                  * ZONE 3
002021                                  *
002022  0820  8054                      MAT7    IO      =$R4,=$R2               OUTPUT DATA TO ZONE 3
        0821  0052
002023  0822  8057                              IO      =$R7,=$R1               OUTPUT ADRS
        0823  0051
002024  0824  8056                              IO      =$R6,=$R3               INPUT DATA (DUMMY FOR ZONE BUMP)
        0825  0053
002025  0826  8056                              IO      =$R6,=$R3               INPUT DATA (DUMMY FOR ZONE BUMP)
        0827  0053
002026  0828  8056                              IO      =$R6,=$R3               INPUT DATA FROM ZONE 3
        0829  0053
002027  082A  C956                              CMR     $R4,=$R6
002028  082B  0905                              BE      >MAT8
002029                                  *
002030  082C  2C03                              LDV     $R2,=3                  PASS ZONE NO. (R2 RESTORED)
002031  082D  B854                              LDR     $R3,=$R4                PASS DATA S.B. (R3 RESTORED)
002032  082E  E380 0970                        LNJ     $B6,<MATERR             **  ERROR "MAT"  **    REPLACE DB IN R3
002033                                  *
002034                                  * ZONE 4
002035                                  *
002036  0830  8054                      MAT8    IO      =$R4,=$R2               OUTPUT DATA TO ZONE 4
        0831  0052
002037  0832  8057                              IO      =$R7,=$R1               OUTPUT ADRS
        0833  0051
002038  0834  8056                              IO      =$R6,=$R3               INPUT DATA (DUMMY FOR ZONE BUMP)
        0835  0053
002039  0836  8056                              IO      =$R6,=$R3               INPUT DATA (DUMMY FOR ZONE BUMP)
        0837  0053
002040  0838  8056                              IO      =$R6,=$R3               INPUT DATA (DUMMY FOR ZONE BUMP)
        0839  0053
002041  083A  8056                              IO      =$R6,=$R3               INPUT DATA FROM ZONE 4
        083B  0053
002042  083C  C956                              CMR     $R4,=$R6
002043  083D  0905                              BE      >MAT9
002044                                  *
002045  083E  2C04                              LDV     $R2,=4                  PASS ZONE NO. (R2 RESTORED)
002046  083F  B854                              LDR     $R3,=$R4                PASS DATA S.B. (R3 RESTORED)
002047  0840  E380 0970                        LNJ     $B6,<MATERR             **  ERROR "MAT"  **    REPLACE DB IN R3
002048                                  *
002049                                  * CHECK IF DONE HERE
002050                                  *
002051  0842  8AD7                      MAT9    INC     =$R7                    BUMP TO NEXT 64-BIT WORD ADRS
002052  0843  8881                              DEC     $B1                     DECREMENT COUNTER
002053  0844  8981                              CMZ     $B1
002054  0845  0981 FFBE                        BNE     MAT5                    DO 1K OR 2K TIMES
002055                                  *
002056                                  *
002057                                  * SETUP FOR NEXT TEST
002058                                  *
002059  0847  F800 1497                        LDR     $R7,<RAMADR             RE-INITIALIZE ADRS OF 1ST RAM LOC
002060  0849  E800 1487                        LDR     $R6,<TMP1               RE-INITIALIZE COUNTER (1K OR 2K)
002061  084B  EF00 1488                        STR     $R6,<TMP2
002062                                  *
002063                                  *****
002064                                  *
002065                                  * COMES HERE 1K OR 2K TIMES (DEPENDING ON NO. RAMS).
002066                                  * VERIFY LAST/WRITE COMPLEMENT/VERIFY, EVERY LOCATION, STARTING AT 1ST
002067                                  * RAM LOC..
002068                                  *
002069                                  *****
002070                                  *
002071                                  * ZONE 1
002072                                  *
002073  084D  8057                      MAT10   IO      =$R7,=$R1               OUTPUT ADRS
        084E  0051
002074  084F  8056                              IO      =$R6,=$R3               INPUT DATA FROM ZONE 1
        0850  0053
002075  0851  C956                              CMR     $R4,=$R6
002076  0852  0905                              BE      >MAT11
002077                                  *
002078  0853  2C01                              LDV     $R2,=1                  PASS ZONE NO. (R2 RESTORED)
002079  0854  B854                              LDR     $R3,=$R4                PASS DATA S.B. (R3 RESTORED)
002080  0855  E380 0970                        LNJ     $B6,<MATERR             **  ERROR "MAT"  **    REPLACE DB IN R3
002081                                  *
002082  0857  8057                      MAT11   IO      =$R7,=$R1               OUTPUT ADRS
        0858  0051
002083  0859  8055                              IO      =$R5,=$R2               OUTPUT DATA TO ZONE 1
        085A  0052
```

```
002084 085B 8057                    IO    =$R7,=$R1          OUTPUT ADRS
       085C 0051
002085 085D 8056                    IO    =$R6,=$R3          INPUT DATA FROM ZONE 1
       085E 0053
002086 085F D956                    CMR   $R5,=$R6
002087 0860 0905                    BE    >MAT12
002088                        *
002089 0861 2C01                    LDV   $R2,=1             PASS ZONE NO. (R2 RESTORED)
002090 0862 B855                    LDR   $R3,=$R5           PASS DATA S.B.(R3 RESTORED)
002091 0863 E380 0970               LNJ   $B6,<MATERR        **  ERROR "MAT"  **   REPLACE DB IN R3
002092                        *
002093                        * ZONE 2
002094                        *
002095 0865 8056                MAT12 IO    =$R6,=$R3          INPUT DATA FROM ZONE 2
       0866 0053
002096 0867 C956                    CMR   $R4,=$R6
002097 0868 0905                    BE    >MAT13
002098                        *
002099 0869 2C02                    LDV   $R2,=2             PASS ZONE NO. (R2 RESTORED)
002100 086A B855                    LDR   $R3,=$R4           PASS DATA S.B.(R3 RESTORED)
002101 086B E380 0970               LNJ   $B6,<MATERR        **  ERROR "MAT"  **   REPLACE DB IN R3
002102                        *
002103 086D 8057                MAT13 IO    =$R7,=$R1          OUTPUT ADRS
       086E 0051
002104 086F 8056                    IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
       0870 0053
002105 0871 8055                    IO    =$R5,=$R2          OUTPUT DATA TO ZONE 2
       0872 0052
002106 0873 8057                    IO    =$R7,=$R1          OUTPUT ADRS
       0874 0051
002107 0875 8056                    IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
       0876 0053
002108 0877 8056                    IO    =$R6,=$R3          INPUT DATA FROM ZONE 2
       0878 0053
002109 0879 D956                    CMR   $R5,=$R6
002110 087A 0905                    BE    >MAT14
002111                        *
002112 087B 2C02                    LDV   $R2,=2             PASS ZONE NO. (R2 RESTORED)
002113 087C B855                    LDR   $R3,=$R5           PASS DATA S.B.(R3 RESTORED)
002114 087D E380 0970               LNJ   $B6,<MATERR        **  ERROR "MAT"  **   REPLACE DB IN R3
002115                        *
002116                        * ZONE 3
002117                        *
002118 087F 8056                MAT14 IO    =$R6,=$R3          INPUT DATA FROM ZONE 3
       0880 0053
002119 0881 C956                    CMR   $R4,=$R6
002120 0882 0905                    BE    >MAT15
002121                        *
002122 0883 2C03                    LDV   $R2,=3             PASS ZONE NO. (R2 RESTORED)
002123 0884 B854                    LDR   $R3,=$R4           PASS DATA S.B.(R3 RESTORED)
002124 0885 E380 0970               LNJ   $B6,<MATERR        **  ERROR "MAT"  **   REPLACE DB IN R3
002125                        *
002126 0887 8057                MAT15 IO    =$R7,=$R1          OUTPUT ADRS
       0888 0051
002127 0889 8056                    IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
       088A 0053
002128 088B 8056                    IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
       088C 0053
002129 088D 8055                    IO    =$R5,=$R2          OUTPUT DATA TO ZONE 3
       088E 0052
002130 088F 8057                    IO    =$R7,=$R1          OUTPUT ADRS
       0890 0051
002131 0891 8056                    IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
       0892 0053
002132 0893 8056                    IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
       0894 0053
002133 0895 8056                    IO    =$R6,=$R3          INPUT DATA FROM ZONE 3
       0896 0053
002134 0897 D956                    CMR   $R5,=$R6
002135 0898 0905                    BE    >MAT16
002136                        *
002137 0899 2C03                    LDV   $R2,=3             PASS ZONE NO. (R2 RESTORED)
002138 089A B855                    LDR   $R3,=$R5           PASS DATA S.B.(R3 RESTORED)
002139 089B E380 0970               LNJ   $B6,<MATERR        **  ERROR "MAT"  **   REPLACE DB IN R3
002140                        *
002141                        * ZONE 4
002142                        *
002143 089D 8056                MAT16 IO    =$R6,=$R3          INPUT DATA FROM ZONE 4
       089E 0053
002144 089F C956                    CMR   $R4,=$R6
002145 08A0 0905                    BE    >MAT17
002146                        *
002147 08A1 2C04                    LDV   $R2,=4             PASS ZONE NO. (R2 RESTORED)
002148 08A2 B854                    LDR   $R3,=$R4           PASS DATA S.B.(R3 RESTORED)
002149 08A3 E380 0970               LNJ   $B6,<MATERR        **  ERROR "MAT"  **   REPLACE DB IN R3
002150                        *
002151 08A5 8057                MAT17 IO    =$R7,=$R1          OUTPUT ADRS
       08A6 0051
002152 08A7 8056                    IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
       08A8 0053
002153 08A9 8056                    IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
       08AA 0053
002154 08AB 8056                    IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
       08AC 0053
002155 08AD 8055                    IO    =$R5,=$R2          OUTPUT DATA TO ZONE 4
       08AE 0052
002156 08AF 8057                    IO    =$R7,=$R1          OUTPUT ADRS
       08B0 0051
002157 08B1 8056                    IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
       08B2 0053
002158 08B3 8056                    IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
       08B4 0053
002159 08B5 8056                    IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
       08B6 0053
002160 08B7 8056                    IO    =$R6,=$R3          INPUT DATA FROM ZONE 4
       08B8 0053
002161 08B9 D956                    CMR   $R5,=$R6
002162 08BA 0905                    BE    >MAT18
002163                        *
002164 08BB 2C04                    LDV   $R2,=4             PASS ZONE NO. (R2 RESTORED)
002165 08BC B855                    LDR   $R3,=$R5           PASS DATA S.B.(R3 RESTORED)
002166 08BD E380 0970               LNJ   $B6,<MATERR        **  ERROR "MAT"  **   REPLACE DB IN R3
002167                        *
```

```
002168                          * CHECK IF DONE HERE
002169                          *
002170  08BF  8AD7             MAT18  INC   =$R7              BUMP TO NEXT 64-BIT WORD ADRS
002171  08C0  8881                    DEC   $B1               DECREMENT COUNTER
002172  08C1  8981                    CMZ   $B1
002173  08C2  0981 FF8A               BNE   MAT10             DO 1K OR 2K TIMES
002174                          *
002175                          *
002176                          * SETUP FOR NEXT TEST
002177                          *
002178  08C4  E800 1496               LDR   $R6,<RAM2FL       IS RAM 2 PRESENT?
002179  08C6  6984                    BNEZ  $R6,>MAT19        BRANCH IF YES
002180  08C7  F870 0BFF               LDR   $R7,=Z'0BFF'      HIGHEST ADRS IN 1ST RAM
002181  08C9  0F83                    B     >MAT20
002182  08CA  F870 0FFF        MAT19  LDR   $R7,=Z'0FFF'      HIGHEST ADRS IN 2ND RAM
002183  08CC  E800 1487        MAT20  LDR   $R6,<TMP1         RE-INITIALIZE COUNTER (1K OR 2K)
002184  08CE  EF00 1488               STR   $R6,<TMP2
002185                          *
002186                          *****
002187                          *
002188                          * COMES HERE 1K OR 2K TIMES (DEPENDING ON NO. RAMS).
002189                          * VERIFY LAST/WRITE COMPLEMENT/VERIFY, EVERY LOCATION, STARTING AT
002190                          * HIGHEST RAM LOC..
002191                          *
002192                          *****
002193                          *
002194                          * ZONE 4
002195                          *
002196  08D0  8057             MAT21  IO    =$R7,=$R1          OUTPUT ADRS
      08D1  0051
002197  08D2  8056                     IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
      08D3  0053
002198  08D4  8056                     IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
      08D5  0053
002199  08D6  8056                     IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
      08D7  0053
002200  08D8  8056                     IO    =$R6,=$R3          INPUT DATA FROM ZONE 4
      08D9  0053
002201  08DA  D956                     CMR   $R5,=$R6
002202  08DB  0905                     BE    >MAT22
002203                          *
002204  08DC  2C04                     LDV   $R2,=4             PASS ZONE NO. (R2 RESTORED)
002205  08DD  B855                     LDR   $R3,=$R5           PASS DATA S.B.(R3 RESTORED)
002206  08DE  E380 0970                LNJ   $B6,<MATERR        **  ERROR "MAT"  **   REPLACE DB IN R3
002207                          *
002208  08E0  8057             MAT22  IO    =$R7,=$R1          OUTPUT ADRS
      08E1  0051
002209  08E2  8056                     IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
      08E3  0053
002210  08E4  8056                     IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
      08E5  0053
002211  08E6  8056                     IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
      08E7  0053
002212  08E8  8054                     IO    =$R4,=$R2          OUTPUT DATA TO ZONE 4
      08E9  0052
002213  08EA  8057                     IO    =$R7,=$R1          OUTPUT ADRS
      08EB  0051
002214  08EC  8056                     IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
      08ED  0053
002215  08EE  8056                     IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
      08EF  0053
002216  08F0  8056                     IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
      08F1  0053
002217  08F2  8056                     IO    =$R6,=$R3          INPUT DATA FROM ZONE 4
      08F3  0053
002218  08F4  C956                     CMR   $R4,=$R6
002219  08F5  0905                     BE    >MAT23
002220                          *
002221  08F6  2C04                     LDV   $R2,=4             PASS ZONE NO. (R2 RESTORED)
002222  08F7  B854                     LDR   $R3,=$R4           PASS DATA S.B.(R3 RESTORED)
002223  08F8  E380 0970                LNJ   $B6,<MATERR        **  ERROR "MAT"  **   REPLACE DB IN R3
002224                          *
002225                          * ZONE 3
002226                          *
002227  08FA  8057             MAT23  IO    =$R7,=$R1          OUTPUT ADRS
      08FB  0051
002228  08FC  8056                     IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
      08FD  0053
002229  08FE  8056                     IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
      08FF  0053
002230  0900  8056                     IO    =$R6,=$R3          INPUT DATA FROM ZONE 3
      0901  0053
002231  0902  D956                     CMR   $R5,=$R6
002232  0903  0905                     BE    >MAT24
002233                          *
002234  0904  2C03                     LDV   $R2,=3             PASS ZONE NO. (R2 RESTORED)
002235  0905  B855                     LDR   $R3,=$R5           PASS DATA S.B.(R3 RESTORED)
002236  0906  E380 0970                LNJ   $B6,<MATERR        **  ERROR "MAT"  **   REPLACE DB IN R3
002237                          *
002238  0908  8057             MAT24  IO    =$R7,=$R1          OUTPUT ADRS
      0909  0051
002239  090A  8056                     IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
      090B  0053
002240  090C  8056                     IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
      090D  0053
002241  090E  8054                     IO    =$R4,=$R2          OUTPUT DATA TO ZONE 3
      090F  0052
002242  0910  8057                     IO    =$R7,=$R1          OUTPUT ADRS
      0911  0051
002243  0912  8056                     IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
      0913  0053
002244  0914  8056                     IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
      0915  0053
002245  0916  8056                     IO    =$R6,=$R3          INPUT DATA FROM ZONE 3
      0917  0053
002246  0918  C956                     CMR   $R4,=$R6
002247  0919  0905                     BE    >MAT25
002248                          *
002249  091A  2C03                     LDV   $R2,=3             PASS ZONE NO. (R2 RESTORED)
002250  091B  B854                     LDR   $R3,=$R4           PASS DATA S.B.(R3 RESTORED)
002251  091C  E380 0970                LNJ   $B6,<MATERR        **  ERROR "MAT"  **   REPLACE DB IN R3
002252                          *
002253                          * ZONE 2
```

```
002254                                  *
002255  091E 8057                       MAT25  IO    =$R7,=$R1          OUTPUT ADRS
        091F 0051
002256  0920 8056                              IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
        0921 0053
002257  0922 8056                              IO    =$R6,=$R3          INPUT DATA FROM ZONE 2
        0923 0053
002258  0924 D956                              CMR   $R5,=$R6
002259  0925 0905                              BE    >MAT26
002260                                  *
002261  0926 2C02                              LDV   $R2,=2             PASS ZONE NO. (R2 RESTORED)
002262  0927 B855                              LDR   $R3,=$R5           PASS DATA S.B.(R3 RESTORED)
002263  0928 E380 0970                         LNJ   $B6,<MATERR        **  ERROR "MAT"  **   REPLACE DB IN R3
002264                                  *
002265  092A 8057                       MAT26  IO    =$R7,=$R1          OUTPUT ADRS
        092B 0051
002266  092C 8056                              IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
        092D 0053
002267  092E 8054                              IO    =$R4,=$R2          OUTPUT DATA TO ZONE 2
        092F 0052
002268  0930 8057                              IO    =$R7,=$R1          OUTPUT ADRS
        0931 0051
002269  0932 8056                              IO    =$R6,=$R3          INPUT DATA (DUMMY FOR ZONE BUMP)
        0933 0053
002270  0934 8056                              IO    =$R6,=$R3          INPUT DATA FROM ZONE 2
        0935 0053
002271  0936 C956                              CMR   $R4,=$R6
002272  0937 0905                              BE    >MAT27
002273                                  *
002274  0938 2C02                              LDV   $R2,=2             PASS ZONE NO. (R2 RESTORED)
002275  0939 B854                              LDR   $R3,=$R4           PASS DATA S.B.(R3 RESTORED)
002276  093A E380 0970                         LNJ   $B6,<MATERR        **  ERROR "MAT"  **   REPLACE DB IN R3
002277                                  *
002278                                  * ZONE 1
002279                                  *
002280  093C 8057                       MAT27  IO    =$R7,=$R1          OUTPUT ADRS
        093D 0051
002281  093E 8056                              IO    =$R6,=$R3          INPUT DATA FROM ZONE 1
        093F 0053
002282  0940 D956                              CMR   $R5,=$R6
002283  0941 0905                              BE    >MAT28
002284                                  *
002285  0942 2C01                              LDV   $R2,=1             PASS ZONE NO. (R2 RESTORED)
002286  0943 B855                              LDR   $R3,=$R5           PASS DATA S.B.(R3 RESTORED)
002287  0944 E380 0970                         LNJ   $B6,<MATERR        **  ERROR "MAT"  **   REPLACE DB IN R3
002288                                  *
002289  0946 8057                       MAT28  IO    =$R7,=$R1          OUTPUT ADRS
        0947 0051
002290  0948 8054                              IO    =$R4,=$R2          OUTPUT DATA TO ZONE 1
        0949 0052
002291  094A 8057                              IO    =$R7,=$R1          OUTPUT ADRS
        094B 0051
002292  094C 8056                              IO    =$R6,=$R3          INPUT DATA FROM ZONE 1
        094D 0053
002293  094E C956                              CMR   $R4,=$R6
002294  094F 0905                              BE    >MAT29
002295                                  *
002296  0950 2C01                              LDV   $R2,=1             PASS ZONE NO. (R2 RESTORED)
002297  0951 B854                              LDR   $R3,=$R4           PASS DATA S.B.(R3 RESTORED)
002298  0952 E380 0970                         LNJ   $B6,<MATERR        **  ERROR "MAT"  **   REPLACE DB IN R3
002299                                  *
002300                                  * CHECK IF DONE HERE
002301                                  *
002302  0954 88D7                       MAT29  DEC   =$R7               DECREMENT TO NEXT 64-BIT WORD ADRS
002303  0955 8881                              DEC   $B1                DECREMENT COUNTER
002304  0956 8981                              CMZ   $B1
002305  0957 0981 FF78                         BNE   MAT2I              DO 1K OR 2K TIMES
002306                                  *
002307                                  *
002308                                  * SET UP FOR 2ND TIME THRU "MAT"
002309                                  *
002310  0959 C870 AAAA                         LDR   $R4,=Z'AAAA'       SWITCH DATA PATTERNS
002311  095B D870 5555                         LDR   $R5,=Z'5555'
002312  095D 8880 1489                         DEC   <TMP3
002313  095F 8980 1489                         CMZ   <TMP3
002314  0961 0980 07FE                         BNE   <MAT4A
002315                                  *
002316                                  *
002317  0963 8054                              IO    =$R4,<INSTAT
        0964 0000 14A6
002318  0966 C900 148D                         CMR   $R4,<MODBIT
002319  0968 0901 0085                         BE    MODCK              IF O.K., GO TO NEXT TEST.
002320                                  *
002321  096A D800 148D                  MAT30  LDR   $R5,<MODBIT         STATUS SB
002322  096C E380 11B5                         LNJ   $B6,<STERR         **  ERROR "STATUS"  ** REPLACE WCS MB
002323  096E 0F81 007F                         B     MODCK              GO TO NEXT TEST
002324                                  *
002325                                  *
002326                                  * ERROR "MAT"
002327                                  *
002328                                  *    CALLING SEQUENCE:
002329                                  *        LNJ   $B6,<MATERR
002330                                  *        R6=DATA IS
002331                                  *        R3=DATA SB (RESTORED TO "<INDAT" BEFORE RETURN)
002332                                  *        R7=LOC
002333                                  *        R2=ZONE (RESTORED TO "<OUTDAT" BEFORE RETURN)
002334                                  *
002335  0970 CF40 007C                  MATERR STR   $R4,MATERB
002336  0972 8054                              IO    =$R4,<INSTAT
        0973 0000 14A6
002337  0975 C900 148D                         CMR   $R4,<MODBIT
002338  0977 09F3                              BNE   >MAT30
002339  0978 C840 0074                         LDR   $R4,MATERB         RESTORE R4
002340                                  *
002341  097A EFC0 0015                         STB   $B6,MATER4
002342  097C BF40 0070                         STR   $R3,MATERB
002343  097E 88C0 0011                         DEC   MATER4+$AF-1       ADJUST REPORTING ADRS
002344  0980 88C0 000F                         DEC   MATER4+$AF-1
002345  0982 B800 0000                  X      LDR   $R3,<ZV$AF
002346  0984 3B83                              BODD  $R3,>MATER1        ZV$AF=1 IF SAF
002347  0985 88C0 000A                         DEC   MATER4+$AF-1       DECREMENT AGAIN IF LAF (ZV$AF=2)
002348                                  *
002349  0987 F970 0C00                  MATER1 CMR   $R7,=Z'0C00'
```

```
002350  0989  0283                        BGE   >MATER2                 
002351  098A  3C01                        LDV   $R3,=1                  DB 1 IN ERROR (FOR NO CONSOLE)
002352  098B  0F82                        B     >MATER3                 
002353  098C  3C02                MATER2  LDV   $R3,=2                  DB 2 IN ERROR (FOR NO CONSOLE)
002354                                    *                             
002355  098D  D380 0000       X    MATER3  LNJ   $B5,<ZV$ER              ERROR "MAI"
002356  098F  0F83                        B     >MATER5                 RETURN FROM ZV$ER
002357  0990  0000                MATER4  RESV  $AF,0                   REPORTING ADRS
002358  0991  1410                        DC    <MMAT                   "MAT"
002359                                    *                             
002360  0992  3B08                MATER5  BEVN  $R3,>MATER6             
002361        0993  FBC0 0003              CALL  ZV$T,MSG4A              PRINT "   * REPLACE DB 1 *"
        0995  D380 0000       X                                         
        0997  0F80                                                      
        0998  1322                                                      
002362  0999  0F87                        B     >MATER7                 
002363                            MATER6  CALL  ZV$T,MSG4B              PRINT "   * REPLACE DB 2 *"
        099A  FBC0 0003                                                 
        099C  D380 0000       X                                         
        099E  0F80                                                      
        099F  132C                                                      
002364                                    *                             
002365                            MATER7  CALL  ZV$TC,MSG19             PRINT "DATA ="
        09A0  FBC0 0003                                                 
        09A2  D380 0000       X                                         
        09A4  0F80                                                      
        09A5  1453                                                      
002366  09A6  EF40 0045                   STR   $R6,MATERA              
002367                                    CALL  ZV$THZ,MATERA           PRINT DATA READ
        09A8  FBC0 0003                                                 
        09AA  D380 0000       X                                         
        09AC  0F80                                                      
        09AD  09EC                                                      
002368                                    *                             
002369                                    CALL  ZV$T,MSG7               PRINT ",  SB"
        09AE  FBC0 0003                                                 
        09B0  D380 0000       X                                         
        09B2  0F80                                                      
        09B3  1415                                                      
002370                                    CALL  ZV$THZ,MATERB           PRINT DATA SB
        09B4  FBC0 0003                                                 
        09B6  D380 0000       X                                         
        09B8  0F80                                                      
        09B9  09ED                                                      
002371                                    *                             
002372                                    CALL  ZV$T,MSG21              PRINT ",  LOC"
        09BA  FBC0 0003                                                 
        09BC  D380 0000       X                                         
        09BE  0F80                                                      
        09BF  1457                                                      
002373  09C0  FF40 002B                   STR   $R7,MATERA              
002374                                    CALL  ZV$THZ,MATERA           PRINT LOC IN ERROR
        09C2  FBC0 0003                                                 
        09C4  D380 0000       X                                         
        09C6  0F80                                                      
        09C7  09EC                                                      
002375                                    *                             
002376                                    CALL  ZV$T,MSG22              PRINT ",  ZONE"
        09C8  FBC0 0003                                                 
        09CA  D380 0000       X                                         
        09CC  0F80                                                      
        09CD  145B                                                      
002377  09CE  AF40 001D                   STR   $R2,MATERA              
002378                                    CALL  ZV$TD,MATERA            PRINT ZONE IN ERROR
        09D0  FBC0 0003                                                 
        09D2  D380 0000       X                                         
        09D4  0F80                                                      
        09D5  09EC                                                      
002379                                    *                             
002380  09D6  8052                        IO    =$R2,<INSTAT            
        09D7  0000 14A6                                                 
002381                                    CALL  ZV$T,MSG23              PRINT ",  STATUS ="
        09D9  FBC0 0003                                                 
        09DB  D380 0000       X                                         
        09DD  0F80                                                      
        09DE  145F                                                      
002382  09DF  AF40 000C                   STR   $R2,MATERA              
002383                                    CALL  ZV$THZ,MATERA           PRINT STATUS READ
        09E1  FBC0 0003                                                 
        09E3  D380 0000       X                                         
        09E5  0F80                                                      
        09E6  09EC                                                      
002384                                    *                             
002385  09E7  A800 14A3                   LDR   $R2,<OUTDAT             RESTORE R2
002386  09E9  B800 14A5                   LDR   $R3,<INDAT              RESTORE R3
002387  09EB  8386                        JMP   $B6                     RETURN
002388                                    *                             
002389  09EC  0000                MATERA  RESV  1,0                     TEMP SAVE
002390  09ED  0000                MATERB  RESV  1,0                     TEMP SAVE
002391                                    *                             
002392                                    *                             
002393                                    *                             
002394                                    *                             
002395                                    *                             
002396  09EE  9870 0100           MODCK   LDR   $R1,=Z'0100'            
002397  09F0  A800 148D                   LDR   $R2,<MODBIT             
002398  09F2  A451                        OR    $R2,=$R1                
002399  09F3  AF40 0A9A                   STR   $R2,STABIT              
002400  09F5  9870 0140                   LDR   $R1,=Z'0140'            
002401  09F7  9900 148E                   CMR   $R1,<STABIT             TRANSPARENT MODE?
002402  09F9  0901 05DF                   BE    TIT10                   YES
002403                                    *                             
002404                                    *                             
002405                                    *                             
002406                                    *                             
002407                                    *  SEQUENTIAL MODE TEST       
002408                                    **********************        
002409                                    *                             
002410                                    *                             
002411                                    *                             
002412  09FB  0F7F                        NOP   >$-1                    
002413                                    *                             
```

```
002414                                  *
002415                                  *  TEST 1
002416                                  *
002417                                  *
002418  09FC  8740 0AA3                        CL    TSTCNT              CLEAR TEST COUNTER.  IT SHOWS WITCH
002419                                  *                                 TEST THAT FAILS
002420                                  *
002421                                  *
002422                                  *
002423                                  *  SPLASH REGISTER AND ADDRESS COUNTER TEST
002424                                  ****************************************
002425                                  *
002426                                  *
002427  09FE  89C0 0A90                 SRT10  CMZ   PASCNT+$AF-1        1ST PASS?
002428  0A00  0990                             BNE   >SRT15               NO
002429  0A01  E380 1225                        LNJ   $B6,<PRCRLF          YES.  PRINT C.R./L.F.
002430                                          CALL  ZV$TC,MSGSP          PRINT MESSAGE
        0A03  FBC0 0003
        0A05  D380 0000       X
        0A07  0F80
        0A08  138C
002431                                          CALL  ZV$TC,MSGSQ
        0A09  FBC0 0003
        0A0B  D380 0000       X
        0A0D  0F80
        0A0E  13AC
002432  0A0F  0004                             RTCN
002433                                  *
002434  0A10  8AC0 0A8F                 SRT15  INC   TSTCNT              BUMP TEST COUNTER
002435                                  *  PRELOAD WCS WITH MICROINSTRUCTION CODES
002436                                  *  FIRST CHECK NO. OF RAMS IN WCS
002437                                  *
002438  0A12  9870 0800                        LDR   $R1,=Z'0800'         SPLASH ADDR FOR RAM1
002439  0A14  9F40 0A86                        STR   $R1,SPHADR
002440  0A16  9870 0080                        LDR   $R1,=Z'0080'
002441  0A18  9F40 0A83                        STR   $R1,SPHCOD           SPLASH MICROINSTRUCTION CODE FOR RAM1
002442  0A1A  C870 0093                 SRT20  LDR   $R4,=Z'0093'         MICROINSTRUCTIONS FOR UNCONDITIONAL
002443  0A1C  D870 C700                        LDR   $R5,=Z'C700'         JUMP TO 16 LOCATIONS AHEAD OF CURRENT
002444  0A1E  E870 2000                        LDR   $R6,=Z'2000'         LOCATION.  THIS MICROINSTRUCTION WILL
002445  0A20  F870 800F                        LDR   $R7,=Z'800F'         BE GENERATED 16 TIMES
002446                                  *
002447  0A22  8000 149B                        IO    <SPHADR,<OUTADR      WCS FIRST ENTRY LOCATION
        0A24  0000 14A2
002448  0A26  7E01                      SRT30  ADV   $R7,=1
002449  0A27  E3C0 01E3                        LNJ   $B6,CSB20            LOAD WCS
002450  0A29  F970 801F                        CMR   $R7,=Z'801F'         ALL 16 ENTRY LOCATIONS LOADED?
002451  0A2B  09FB                             BNE   >SRT30               NO
002452                                  *
002453  0A2C  1CF1                              LDV   $R1,=-15             COUNTER
002454                                  *
002455  0A2D  ABC0 01C0                         LAB   $B2,MC23             FETCH MICROCODE INC R1.  AND
002456  0A2F  E3C0 01D2                         LNJ   $B6,CSB10            FILL IN 16 CONSECUTIVE LOCATIONS
002457  0A31  E3C0 01D9                 SRT40  LNJ   $B6,CSB20            WRITE WCS
002458  0A33  17FE                             BINC  $R1,>SRT40           BRANCH IF ALL 15 LOCATIONS NOT LOADED
002459                                  *
002460  0A34  89C0 0A61                         CMZ   RAM2FL               RAM2 PRESENT?
002461  0A36  0981 0029                         BNE   SRT90                YES
002462  0A38  F870 83FC                         LDR   $R7,=Z'83FC'         LOAD LAST 16TH MICROWORD FOR INC. R1
002463  0A3A  E3C0 01D0                         LNJ   $B6,CSB20
002464  0A3C  F870 0BFC                         LDR   $R7,=Z'0BFC'
002465  0A3E  ABC0 0A69                         LAB   $B2,BUFF1            STORES RETURN ROUTINE FOR RAM1 ONLY
002466  0A40  9870 0BFF                         LDR   $R1,=Z'0BFF'
002467  0A42  9F40 0A5B                         STR   $R1,WCSIZE           WCS LAST LOCATION
002468  0A44  8057                      SRT45  IO    =$R7,<OUTADR          WCS ADDR FOR RETURN ROUTINE
        0A45  0000 14A2
002469  0A47  6CF0                             LDV   $R6,=-16             TO COUNT 16 LOCATIONS IN RETURN ROUTINE
002470  0A48  8072                      SRT50  IO    +$B2,<OUTDAT
        0A49  0000 14A3
002471  0A4B  67FD                             BINC  $R6,>SRT50           ALL LOCATIONS LOADED?
002472  0A4C  0F83                             B     >SRT55
002473  0A4D  E3C0 07E9                         LNJ   $B6,RDBK10           JUMP TO SUBROUTINE TO READ WCS
002474                                  *                                  BACK TO A BUFFER FOR DEBUG PURPOSE
002475                                  *
002476                                  *
002477                                  *  STARTS SPLASH TEST HERE
002478                                  *
002479  0A4F  2C10                      SRT55  LDV   $R2,=16               INITIALIZE COUNTER TO 16
002480  0A50  8840 0A4B                        LDR   $R3,SPHCOD           SPLASH MICROCODE
002481  0A52  BF40 0004                 SRT60  STR   $R3,SRT70
002482                                  *
002483  0A54  E3C0 0042                        LNJ   $B6,SRT210           ENABLE WCS AND STATUS CHECK
002484                                  *
002485                                  *
002486  0A56  8751                             CL    =$R1
002487                                  *
002488  0A57  0000                      SRT70  RESV  1,0                   SPLASH.  ERROR IF O.K. NOT PRINTED
002489                                  *                                  IN 5 SEC.
002490                                  *
002491  0A58  9952                             CMR   $R1,=$R2             RETURN FROM WCS FIRMWARE.  SPLASH ERROR?
002492  0A59  0981 002D                         BNE   SRT200               ERROR.  BRANCH TO REPORT ERROR
002493  0A5B  88D2                      SRT80  DEC   =$R2                 COUNT 1
002494  0A5C  8AD3                             INC   =$R3                 BUMP SPLASH MICROCODE (R3) FOR NEXT SPLASH
002495  0A5D  2D00                             CMV   $R2,=0               ALL 16 SPLASH TESTED?
002496  0A5E  0374                             BG    >SRT60               NO.  CONTINUE FOR NEXT SPLASH TEST
002497  0A5F  0F8F                             B     >SRT100              YES.  ALL 16 SPLASH TESTED
002498                                  *
002499                                  *
002500  0A60  F870 87FC                 SRT90  LDR   $R7,=Z'87FC'         LOAD LAST 16TH MICROWORD FOR INC. R1
002501  0A62  E3C0 01A8                        LNJ   $B6,CSB20
002502  0A64  F870 0FFC                        LDR   $R7,=Z'0FFC'
002503  0A66  ABC0 0A51                        LAB   $B2,BUFF2            STORES RETURN ROUTINE WHEN RAM2 PRESENT
002504  0A68  9B70 0FFF                        LDR   $R1,=Z'0FFF'         WCS LAST LOCATION
002505  0A6A  9F40 0A33                        STR   $R1,WCSIZE
002506  0A6C  0F81 FFD7                        B     SRT45
002507                                  *
002508                                  *
002509                                  *
002510                                  *
002511  0A6E  89C0 0A27                 SRT100 CMZ   RAM2FL               RAM2 PRESENT?
002512  0A70  0901 005E                        BE    ACT10                NO.  RAM1 ONLY.  DONE
002513  0A72  0F81 005C                        B     ACT10
002514                                  *
002515                                  *
```

```
002516                                  *
002517                                  *
002518  0A74  9870 0F00                        LDR   $R1,=Z'0F00'      SPLASH  ADDR FOR  RAM2
002519  0A76  9F40 0A24                        STR   $R1,SPHADR
002520  0A78  9870 00C0                        LDR   $R1,=Z'00C0'      SPLASH MICROCODE
002521  0A7A  9F40 0A21                        STR   $R1,SPHCOD
002522  0A7C  8751                            CL    =$R1              CLEAR RAM1 16 ENTRY
002523  0A7D  2CF0                            LDV   $R2,=-16          LOCATIONS
002524  0A7E  8000 149B                        IO    <SPHADR,<OUTADR
        0A80  0000 14A2
002525  0A82  8051                    SRT110  IO    =$R1,<OUTDAT
        0A83  0000 14A3
002526  0A85  27FD                            BINC  $R2,>SRT110
002527  0A86  0000                            HLT
002528                                  *
002529                                  *
002530                                  SRT200  CALL  ZV$ER,SR1,MSGSR1  ERROR "SR1"
        0A87  FBC0 0003
        0A89  D380 0000         X
        0A8B  0F80
        0A8C  0A94
        0A8D  146D
002531                                          CALL  ZV$T,MSG4         PRINT "  * REPLACE MB *"
        0A8E  FBC0 0003
        0A90  D380 0000         X
        0A92  0F80
        0A93  1319
002532  0A94  0F7F                      SR1     NOP   >$-1              **  ERROR  **
002533                                  *                             SPLASH ERROR
002534  0A95  0F81 FFC5                         B     SRT80             CONTINUE NEXT SPLASH TEST
002535                                  *
002536                                  *
002537                                  *
002538                                  *  ENABLE WCS AND STATUS CHECK
002539                                  *
002540  0A97  8070 0800                 SRT210  IO    =Z'0800',<OUTCON  ENABLE WCS
        0A99  0000 14A1
002541  0A9B  AF40 09EC                         STR   $R2,TMP2          SAVE R2
002542  0A9D  8052                             IO    =$R2,<INSTAT
        0A9E  0000 14A6
002543  0AA0  A900 148E                         CMR   $R2,<STABIT
002544  0AA2  0901 0029                         BE    SRT220
002545                                          CALL  ZV$ER,ST1,MSGST1  ERROR "ST1"
        0AA4  FBC0 0003
        0AA6  D380 0000         X
        0AA8  0F80
        0AA9  0AC9
        0AAA  1473
002546                                          CALL  ZV$T,MSG23        PRINT ",   STATUS ="
        0AAB  FBC0 0003
        0AAD  D380 0000         X
        0AAF  0F80
        0AB0  145F
002547  0AB1  AF40 09D7                         STR   $R2,TMP3
002548                                          CALL  ZV$THZ,TMP3       PRINT STATUS READ
        0AB3  FBC0 0003
        0AB5  D380 0000         X
        0AB7  0F80
        0AB8  1489
002549  0AB9  DF40 09CD                         STR   $R5,TMP1          SAVE R5
002550  0ABB  D840 09E4                         LDR   $R5,TSTCNT        R5 SHOWS WHICH TEST THAT FAILS
002551                                          CALL  ZV$T,MSG24        PRINT ",   TEST NO. ="
        0ABD  FBC0 0003
        0ABF  D380 0000         X
        0AC1  0F80
        0AC2  1465
002552                                          CALL  ZV$THZ,TSTCNT     PRINT TEST NO.
        0AC3  FBC0 0003
        0AC5  D380 0000         X
        0AC7  0F80
        0AC8  14A0
002553  0AC9  0F7F                      ST1     NOP   >$-1              **  ERROR  **
002554                                  *                             STATUS ERROR (IN R2)
002555  0ACA  D840 09BC                         LDR   $R5,TMP1          RESTORE R5
002556  0ACC  A840 09BB                 SRT220  LDR   $R2,TMP2          RESTORE R2
002557                                  *
002558                                  *
002559  0ACE  8386                             JMP   $B6               RETURN
002560                                  *
002561                                  *
002562                                  *
002563                                  *
002564                                  *
002565                                  *
002566                                  *  TEST 2
002567                                  *
002568                                  *
002569                                  *  NOW STARTS ADDRESS COUNTER TEST
002570                                  *  PRELOAD WCS WITH MICROCODES
002571                                  *
002572                                  *
002573  0ACF  0F7F                      ACT10   NOP   >$-1
002574                                  *
002575                                  *
002576  0AD0  9870 0800                         LDR   $R1,=Z'0800'      SPLASH ADDR FOR RAM1
002577  0AD2  9F40 09C8                         STR   $R1,SPHADR
002578  0AD4  9870 0080                         LDR   $R1,=Z'0080'      SPLASH MICROCODE  FOR RAM1
002579  0AD6  9F40 09C5                         STR   $R1,SPHCOD
002580                                  *
002581                                  *
002582                                  *
002583  0AD8  A840 09C5                         LDR   $R2,WCSIZE        LAST LOCATION IN WCS (CALCULATED
002584                                  *                               IN SPLASH REGISTER TEST)
002585  0ADA  2EFC                              ADV   $R2,=-4           LAST LOCATION FOR TESTING
002586                                  *
002587  0ADB  8AC0 09C4                         INC   TSTCNT            BUMP TEST COUNTER
002588  0ADD  8000 149B                         IO    <SPHADR,<OUTADR   WCS FIRST ENTRY LOCATION
        0ADF  0000 14A2
002589  0AE1  9840 09B9                         LDR   $R1,SPHADR        SPLASH ADDR.  R1 = COUNTER
002590  0AE3  ABC0 010A                         LAB   $B2,MC23          FETCH MICROCODE FOR INC R1.  AND
002591  0AE5  E3C0 011C                         LNJ   $B6,CSB10         FILL THE RAM WITH INC R1
002592  0AE7  E3C0 0123                 ACT20   LNJ   $B6,CSB20         LOAD WCS
002593  0AE9  1E01                             ADV   $R1,=1
```

```
002594  0AEA  9952              CMR   $R1,=$R2          ALL LOCATIONS IN WCS LOADED?
002595  0AEB  03FC              BLE   >ACT20            NO
002596  0AEC  0F83              B     >ACT30
002597                      *
002598  0AED  E3C0 0749         LNJ   $B6,RDBK10        YES.  JUMP TO SUBROUTINE TO READ WCS
002599                      *                           BACK TO MEMORY LOW FOR DEBUG PURPOSE
002600                      *
002601                      *  STARTS TEST HERE
002602                      *
002603  0AEF  E3C0 FFA7     ACT30 LNJ   $B6,SRT210        ENABLE WCS AND STATUS CHECK
002604                      *
002605                      *
002606                      *
002607  0AF1  B840 09AA          LDR   $R3,SPHCOD        SPLASH MICROCODE
002608  0AF3  BF40 0005          STR   $R3,ACT40
002609  0AF5  9870 0800          LDR   $R1,=Z'0800'      INITIALIZE COUNTER TO 800
002610  0AF7  0F81 0001          B     ACT40
002611                      *
002612  0AF9  0000          ACT40 RESV  1,0               SPLASH.  ERROR IF "OK" NOT PRINTED
002613                      *                           IN 5 SEC.
002614  0AFA  A840 09A3          LDR   $R2,WCSIZE        RETURN FROM WCS FIRMWARE
002615  0AFC  2EFD               ADV   $R2,=-3           R2 NOW SHOULD BE THE NO. OF COUNTS IN R1
002616  0AFD  9952              CMR   $R1,=$R2          ERROR?
002617  0AFE  0982              BNE   >ACT50            YES.  REPORT ERROR
002618  0AFF  0F8F              B     >RAT10            GO TO NEXT TEST
002619                      *
002620                      ACT50 CALL  ZV$ER,AC1,MSGAC1  ERROR "AC1"
        0B00  FBC0 0003
        0B02  D380 0000      X
        0B04  0F80
        0B05  0B0D
        0B06  146F
002621                            CALL  ZV$T,MSG4       PRINT "  * REPLACE MB *"
        0B07  FBC0 0003
        0B09  D380 0000      X
        0B0B  0F80
        0B0C  1319
002622  0B0D  0F7F          AC1   NOP   >$-1            **  ERROR  **
002623                      *                           ADDRESS COUNTER ERROR
002624                      *                           GO TO NEXT TEST
002625                      *
002626                      *
002627                      *
002628                      *
002629                      *
002630                      *
002631                      *
002632                      *  TEST 3
002633                      *
002634                      *
002635                      *  RETURN ADDRESS REGISTER AND JUMP ADDRESS REGISTER TEST
002636                      ***************************************************
002637                      *
002638                      *
002639                      *
002640  0B0E  0F7F          RAT10 NOP   >$-1
002641                      *
002642                      *
002643  0B0F  9840 098E          LDR   $R1,WCSIZE        LAST LOCATION IN WCS
002644  0B11  1EFC               ADV   $R1,=-4           LOCATION FOR SUBROUTINE
002645  0B12  8AC0 098D          INC   TSTCNT            BUMP TEST COUNTER
002646                      *
002647                      *
002648  0B14  C870 11A8          LDR   $R4,=Z'11A8'      MICROCODE FOR SUBROUTINE (INCR  D1,D1;
002649  0B16  D870 C30A          LDR   $R5,=Z'C30A'      RETURN)
002650  0B18  E870 2000          LDR   $R6,=Z'2000'
002651  0B1A  F870 27FF          LDR   $R7,=Z'27FF'
002652                      *
002653  0B1C  8051               IO    =$R1,<OUTADR      ADDR FOR SUBROUTINE
        0B1D  0000 14A2
002654  0B1F  E3C0 00EB          LNJ   $B6,CSB20         LOAD SUBROUTINE MICROCODE TO WCS
002655                      *
002656                      *
002657  0B21  F870 8002          LDR   $R7,=Z'8002'      INITIAL VALUE TO CALCULATE SUBROUTINE
002658  0B23  FF40 0968          STR   $R7,STOR1         CALL ADDR IN WCS
002659                      *
002660  0B25  C870 0093     RAT20 LDR   $R4,=Z'0093'      MICROCODE FOR UNCONDITIONAL JUMP TO
002661  0B27  D870 C700          LDR   $R5,=Z'C700'      SUBROUTINE CALL
002662  0B29  E870 2000          LDR   $R6,=Z'2000'
002663                      *
002664  0B2B  8000 149B          IO    <SPHADR,<OUTADR   WCS FIRST ENTRY LOCATION
        0B2D  0000 14A2
002665  0B2F  8054               IO    =$R4,<OUTDAT
        0B30  0000 14A3
002666  0B32  8055               IO    =$R5,<OUTDAT
        0B33  0000 14A3
002667  0B35  8056               IO    =$R6,<OUTDAT
        0B36  0000 14A3
002668                      *
002669  0B38  F840 0953          LDR   $R7,STOR1         CALCULATE THE LOCATION WHERE THE
002670  0B3A  7E01               ADV   $R7,=1            UNCONDITIONAL JUMP IS GOING TO FOR
002671  0B3B  FF40 0950          STR   $R7,STOR1         SUBROUTINE CALL (8003 1ST TIME, 8004
002672                      *                           2ND TIME, 8005 3RD TIME ......)
002673  0B3D  8057               IO    =$R7,<OUTDAT      LAST LOCATION FOR MICROCODE  "GOTO"
        0B3E  0000 14A3
002674  0B40  FA70 8800          ADD   $R7,=Z'8800'
002675  0B42  8057               IO    =$R7,<OUTADR      ADDR FOR STORING SUBROUTINE CALL
        0B43  0000 14A2
002676                      *
002677  0B45  C870 0093          LDR   $R4,=Z'0093'      MICROCODE FOR SUBROUTINE CALL
002678  0B47  D870 C300          LDR   $R5,=Z'C300'
002679  0B49  E870 2000          LDR   $R6,=Z'2000'
002680                      *
002681  0B4B  89C0 094A          CMZ   RAM2FL            RAM2  PRESENT?
002682  0B4D  0981 002A          BNE   RAT100            YES
002683  0B4F  F870 C3FB          LDR   $R7,=Z'C3FB'      NO.  LAST 16 BIT WORD MICROCODE FOR
002684                      *                           SUBROUTINE CALL
002685  0B51  E3C0 00B9     RAT40 LNJ   $B6,CSB20         LOAD MICROCODE FOR SUBROUTINE CALL
002686                      *
002687  0B53  C870 0093          LDR   $R4,=Z'0093'      MICROCODE FOR UNCONDITIONAL JUMP TO
002688  0B55  D870 C700          LDR   $R5,=Z'C700'      RETURN ROUTINE
002689  0B57  E870 2000          LDR   $R6,=Z'2000'
002690  0B59  89C0 093C          CMZ   RAM2FL            RAM2 PRESENT?
```

```
002691  0B5B  0981 0020          BNE   RAT110              YES
002692  0B5D  F870 83FC          LDR   $R7,=Z'83FC'        NO.  LAST 16 BIT MICROWORD FOR
002693                           *                         UNCONDITIONAL JUMP TO RETURN ROUTINE
002694  0B5F  E3C0 00AB          RAT50 LNJ   $B6,CSB20       LOAD MICROCODE FOR UNCONDITIONAL JUMP
002695                           *                         TO RETURN ROUTINE
002696                           *
002697                           *  STARTS TEST HERE
002698                           *
002699  0B61  9840 093A          LDR   $R1,SPHCOD          SPLASH MICROCODE
002700  0B63  9F40 0004          STR   $R1,RAT60
002701                           *
002702                           *
002703  0B65  E3C0 FF31          LNJ   $B6,SRT210          ENABLE WCS AND STATUS CHECK
002704                           *
002705                           *
002706  0B67  8751               CL    =$R1
002707                           *
002708  0B68  0000               RAT60 RESV  1,0           SPLASH.  ERROR IF O.K. NOT PRINTED
002709                           *                         IN 5 SEC
002710  0B69  1D01               CMV   $R1,=1              ERROR?
002711  0B6A  0981 001B          BNE   RAT200              YES.  REPORT ERROR
002712  0B6C  A840 091F          LDR   $R2,STOR1           NO
002713  0B6E  89C0 0927          CMZ   RAM2FL              RAM2 PRESENT?
002714  0B70  0981 000F          BNE   RAT120              YES
002715  0B72  A970 83F7          CMR   $R2,=Z'83F7'        NO.  ALL REGISTER BITS TESTED?
002716  0B74  0381 FFB0          BLE   RAT20               NO.  TEST NEXT BIT
002717  0B76  0F81 00A1          B     CBT10               YES.  GO TO NEXT TEST
002718                           *
002719                           *
002720  0B78  F870 C7FB          RAT100 LDR  $R7,=Z'C7FB'  LAST 16 BIT MICROWORD FOR
002721  0B7A  0F81 FFD6          B     RAT40               SUBROUTINE CALL
002722                           *
002723  0B7C  F870 87FC          RAT110 LDR  $R7,=Z'87FC'  LAST 16 BIT MICROWORD FOR
002724  0B7E  0F81 FFE0          B     RAT50               UNCONDITIONAL JUMP TO RETURN ROUTINE
002725                           *
002726  0B80  A970 87F7          RAT120 CMR  $R2,=Z'87F7'  ALL REGISTER BITS TESTED?
002727  0B82  0381 FFA2          BLE   RAT20               NO.  TEST NEXT BIT
002728  0B84  0F81 0093          B     CBT10               YES.  GO TO NEXT TEST
002729                           *
002730                           *
002731                           RAT200 CALL ZV$ER,RA1,MSGRA1  ERROR "RA1"
        0B86  FBC0 0003
        0B88  D380 0000      X
        0B8A  0F80
        0B8B  0B93
        0B8C  1471
002732                                 CALL  ZV$T,MSG4      PRINT "  * REPLACE MB *"
        0B8D  FBC0 0003
        0B8F  D380 0000      X
        0B91  0F80
        0B92  1319
002733  0B93  0F7F               RA1   NOP   >$-1            **  ERROR  **
002734  0B94  0F81 0083          B     CBT10               GO TO NEXT TEST
002735                           *
002736                           *
002737                           *
002738                           *  MICROCODES FOR SEQUENTIAL MODE TESTS
002739                           ************************************
002740                           *
002741                           *
002742  0B96  0093               MC1   DC    Z'0093'       SET ZERO
002743  0B97  CF00                     DC    Z'CF00'       (FLOPS  ZR1)
002744  0B98  25C0                     DC    Z'25C0'
002745  0B99  07FF                     DC    Z'07FF'
002746                           *
002747  0B9A  0093               MC2   DC    Z'0093'       CLEAR ZERO
002748  0B9B  CF00                     DC    Z'CF00'       (FLOPS  ZR0)
002749  0B9C  2580                     DC    Z'2580'
002750  0B9D  07FF                     DC    Z'07FF'
002751                           *
002752  0B9E  0093               MC3   DC    Z'0093'       BRANCH TO 0BFB (RAM1 ONLY) IF FALSE
002753  0B9F  CF00                     DC    Z'CF00'       (IFZERO  ,RAM1)
002754  0BA0  2033                     DC    Z'2033'
002755  0BA1  83FB                     DC    Z'83FB'
002756                           *
002757  0BA2  0093               MC4   DC    Z'0093'       BRANCH TO 0FFB (RAM2 PRESENT) IF FALSE
002758  0BA3  CF00                     DC    Z'CF00'       (IFZERO  ,RAM2)
002759  0BA4  2033                     DC    Z'2033'
002760  0BA5  87FB                     DC    Z'87FB'
002761                           *
002762  0BA6  0093               MC5   DC    Z'0093'       BRANCH TO 0BFB (RAM1 ONLY) IF TRUE
002763  0BA7  CF00                     DC    Z'CF00'       (IFZERO  ,RAM1)
002764  0BA8  2033                     DC    Z'2033'
002765  0BA9  03FB                     DC    Z'03FB'
002766                           *
002767  0BAA  0093               MC6   DC    Z'0093'       BRANCH TO 0FFB (RAM2 PRESENT) IF TRUE
002768  0BAB  CF00                     DC    Z'CF00'       (IFZERO  RAM2)
002769  0BAC  2033                     DC    Z'2033'
002770  0BAD  07FB                     DC    Z'07FB'
002771                           *
002772  0BAE  11A8               MC7   DC    Z'11A8'       INC R1, GO TO RETURN ROUTINE (0BFC
002773  0BAF  C30A                     DC    Z'C30A'       WHEN RAM1 ONLY)
002774  0BB0  2000                     DC    Z'2000'       (INCR  D1;D1  GOTO  RET1)
002775  0BB1  83FC                     DC    Z'83FC'
002776                           *
002777  0BB2  11A8               MC8   DC    Z'11A8'       INC R1, GO TO RETURN ROUTINE (0FFC
002778  0BB3  C30A                     DC    Z'C30A'       WHEN RAM2 PRESENT)
002779  0BB4  2000                     DC    Z'2000'       (INCR  D1;D1  GOTO  RET2)
002780  0BB5  87FC                     DC    Z'87FC'
002781                           *
002782  0BB6  11A8               MC9   DC    Z'11A8'       INC R2
002783  0BB7  C306                     DC    Z'C306'       (INCR  D2)
002784  0BB8  2000                     DC    Z'2000'
002785  0BB9  07FF                     DC    Z'07FF'
002786                           *
002787  0BBA  0093               MC10  DC    Z'0093'       CALL (0BFB WHEN RAM1 ONLY) IF FALSE
002788  0BBB  CB00                     DC    Z'CB00'       (IFZERO  ,RAM1,CALL)
002789  0BBC  2033                     DC    Z'2033'
002790  0BBD  C3FB                     DC    Z'C3FB'
002791                           *
002792  0BBE  0093               MC11  DC    Z'0093'       CALL (0FFB WHEN RAM2 PRESENT) IF FALSE
002793  0BBF  CB00                     DC    Z'CB00'       (IFZERO  ,RAM2,CALL)
002794  0BC0  2033                     DC    Z'2033'
```

```
002795  0BC1  C7FB                    DC    2'C7FB'
002796                        *
002797  0BC2  0093            MC12    DC    2'0093'          CALL (0BFB WHEN RAM1 ONLY) IF TRUE
002798  0BC3  CB00                    DC    2'CB00'          (IFZERO  RAM1,,CALL)
002799  0BC4  2033                    DC    2'2033'
002800  0BC5  43FB                    DC    2'43FB'
002801                        *
002802  0BC6  0093            MC13    DC    2'0093'          CALL (0FFB WHEN RAM2 PRESENT) IF TRUE
002803  0BC7  CB00                    DC    2'CB00'          (IFZERO  RAM2,,CALL)
002804  0BC8  2033                    DC    2'2033'
002805  0BC9  47FB                    DC    2'47FB'
002806                        *
002807  0BCA  11A8            MC14    DC    2'11A8'          INC R2, RETURN
002808  0BCB  C306                    DC    2'C306'          (INCR  D2;D2  RETURN)
002809  0BCC  2000                    DC    2'2000'
002810  0BCD  27FF                    DC    2'27FF'
002811                        *
002812  0BCE  0093            MC15    DC    2'0093'          CALL (0BFB WHEN RAM1 ONLY)
002813  0BCF  CB00                    DC    2'CB00'          (CALL  RAM1)
002814  0BD0  2000                    DC    2'2000'
002815  0BD1  C3FB                    DC    2'C3FB'
002816                        *
002817  0BD2  0093            MC16    DC    2'0093'          CALL (0FFB WHEN RAM2 PRESENT)
002818  0BD3  CB00                    DC    2'CB00'          (CALL  RAM2)
002819  0BD4  2000                    DC    2'2000'
002820  0BD5  C7FB                    DC    2'C7FB'
002821                        *
002822  0BD6  0093            MC17    DC    2'0093'          RETURN IF TRUE OTHERWISE GO TO RETURN
002823  0BD7  CB00                    DC    2'CB00'          ROUTINE (0BFC WHEN RAM1 ONLY)
002824  0BD8  2033                    DC    2'2033'          (IFZERO  RETURN  ,*XXX)  XXX IS THE
002825  0BD9  A3FC                    DC    2'A3FC'          LOCATION RETURN TO
002826                        *
002827  0BDA  0093            MC18    DC    2'0093'          RETURN IF TRUE OTHERWISE GO TO RETURN
002828  0BDB  CB00                    DC    2'CB00'          ROUTINE (0FFC WHEN RAM2 PRESENT)
002829  0BDC  2033                    DC    2'2033'          (IFZERO  RETURN  ,*YYY)  YYY IS THE
002830  0BDD  A7FC                    DC    2'A7FC'          LOCATION RETURN TO
002831                        *
002832  0BDE  0093            MC19    DC    2'0093'          RETURN IF FALSE OTHERWISE GO TO RETURN
002833  0BDF  CB00                    DC    2'CB00'          ROUTINE (0BFC WHEN RAM1 ONLY)
002834  0BE0  2033                    DC    2'2033'          (IFZERO  *XXX, RETURN)   XXX IS THE
002835  0BE1  23FC                    DC    2'23FC'          LOCATION RETURN TO
002836                        *
002837  0BE2  0093            MC20    DC    2'0093'          RETURN IF FALSE OTHERWISE GO TO RETURN
002838  0BE3  CB00                    DC    2'CB00'          ROUTINE (0FFC WHEN RAM2 PRESENT)
002839  0BE4  2033                    DC    2'2033'          (IFZERO  *YYY, RETURN)   YYY IS THE
002840  0BE5  27FC                    DC    2'27FC'          LOCATION RETURN TO
002841                        *
002842  0BE6  11B8            MC21    DC    2'11B8'          INC R1 AND PUT RESULT IN LINK REGISTER
002843  0BE7  C30A                    DC    2'C30A'
002844  0BE8  2D40                    DC    2'2D40'
002845  0BE9  07FF                    DC    2'07FF'
002846                        *
002847  0BEA  0093            MC22    DC    2'0093'          L - BRANCH
002848  0BEB  CB00                    DC    2'CB00'
002849  0BEC  2000                    DC    2'2000'
002850  0BED  F0FF                    DC    2'F0FF'
002851                        *
002852  0BEE  11A8            MC23    DC    2'11A8'          INC R1
002853  0BEF  C30A                    DC    2'C30A'
002854  0BF0  2000                    DC    2'2000'
002855  0BF1  07FF                    DC    2'07FF'
002856                        *
002857  0BF2  8013            MC24    DC    2'8013'          SET SEL REGISTER
002858  0BF3  CD10                    DC    2'CD10'
002859  0BF4  2A80                    DC    2'2A80'
002860  0BF5  07FF                    DC    2'07FF'
002861                        *
002862  0BF6  0093            MC25    DC    2'0093'          BRANCH TO 0BFB (RAM1 ONLY)
002863  0BF7  CF00                    DC    2'CF00'          IF TRUE FOR SEL REG SET
002864  0BF8  202F                    DC    2'202F'
002865  0BF9  03FB                    DC    2'03FB'
002866                        *
002867  0BFA  0093            MC26    DC    2'0093'          BRANCH TO 0FFB (RAM2 PRESENT)
002868  0BFB  CF00                    DC    2'CF00'          IF TRUE FOR SEL REG SET
002869  0BFC  202F                    DC    2'202F'
002870  0BFD  07FB                    DC    2'07FB'
002871                        *
002872  0BFE  77A8            MC27    DC    2'77A8'          INC B7
002873  0BFF  C300                    DC    2'C300'
002874  0C00  2000                    DC    2'2000'
002875  0C01  07FF                    DC    2'07FF'
002876                        *
002877                        *
002878                        *
002879                        *
002880  0C02  C842 0000       CSB10   LDR   $R4,$B2.0        SUBROUTINE TO PUT MICROCODE IN
002881  0C04  D842 0001               LDR   $R5,$B2.1        R REGISTER
002882  0C06  E842 0002               LDR   $R6,$B2.2
002883  0C08  F842 0003               LDR   $R7,$B2.3
002884  0C0A  8386                    JMP   $B6              RETURN
002885                        *
002886  0C0B  8054            CSB20   IO    =$R4,<OUTDAT     SUBROUTINE TO LOAD MICROCODE INTO WCS
        0C0C  0000 14A3
002887  0C0E  8055                    IO    =$R5,<OUTDAT
        0C0F  0000 14A3
002888  0C11  8056                    IO    =$R6,<OUTDAT
        0C12  0000 14A3
002889  0C14  8057                    IO    =$R7,<OUTDAT
        0C15  0000 14A3
002890  0C17  8386                    JMP   $B6              RETURN
002891                        *
002892                        *
002893                        *
002894                        ************************************************
002895                        *
002896                        *
002897                        *  TEST 4
002898                        *
002899                        *
002900                        *  CONDITIONAL BRANCH TEST
002901                        ************************
002902                        *
002903                        *
```

```
002904                                *  THE TEST CONDITION IS ZERO FLOP SET
002905                                *
002906  0C18  0F7F                   CBT10  NOP    >$-1
002907                                *
002908                                *
002909  0C19  8000 149B                      IO     <SPHADR,<OUTADR       WCS 1ST SPLASH LOCATION
        0C1B  0000 14A2
002910  0C1D  8AC0 0882                      INC    TSTCNT                BUMP TEST COUNTER
002911  0C1F  ABC0 FF76                      LAB    $B2,MC1               FETCH MICROCODE SET ZERO FROM MAIN
002912  0C21  E3C0 FFE0                      LNJ    $B6,CSB10             MEMORY AND PUT IT INTO WCS
002913  0C23  E3C0 FFE7                      LNJ    $B6,CSB20
002914                                *
002915  0C25  89C0 0870                      CMZ    RAM2FL                RAM2 PRESENT?
002916  0C27  0981 0074                      BNE    CBT150                YES
002917  0C29  ABC0 FF74                      LAB    $B2,MC3               NO.  FETCH MICROCODE BRANCH TO 0BFB IF
002918  0C2B  E3C0 FFD6                      LNJ    $B6,CSB10             FALSE FROM MAIN MEMORY AND PUT IT INTO
002919  0C2D  E3C0 FFDD                      LNJ    $B6,CSB20             WCS
002920                                *
002921  0C2F  ABC0 FF7E                      LAB    $B2,MC7               FETCH MICROCODE INC R1 AND GO TO 0BFC
002922  0C31  E3C0 FFD0                      LNJ    $B6,CSB10             (RETURN ROUTINE)
002923  0C33  E3C0 FFD7                      LNJ    $B6,CSB20
002924                                *
002925  0C35  B840 0868               CBT20  LDR    $R3,WCSIZE            LAST LOCATION IN WCS
002926  0C37  3EFC                           ADV    $R3,=-4               LOCATION FOR MICROCODE INC R2
002927  0C38  8053                           IO     =$R3,<OUTADR
        0C39  0000 14A2
002928  0C3B  ABC0 FF7A                      LAB    $B2,MC9               FETCH MICROCODE INC R2 FROM MAIN
002929  0C3D  E3C0 FFC4                      LNJ    $B6,CSB10             MEMORY AND PUT IT INTO WCS
002930  0C3F  E3C0 FFCB                      LNJ    $B6,CSB20
002931                                *
002932                                *
002933                                *
002934                                *  START TEST HERE
002935                                *
002936  0C41  8740 084A                      CL     STOR1                 COUNTER FOR CHECKING ALL TEST
002937                                *                                   CONDITIONS ARE TESTED
002938  0C43  9840 0858               CBT30  LDR    $R1,SPHCOD            SPLASH MICROCODE
002939  0C45  9F40 0005                      STR    $R1,CBT40
002940                                *
002941                                *
002942  0C47  E3C0 FE4F                      LNJ    $B6,SRT210            ENABLE WCS AND STATUS CHECK
002943                                *
002944  0C49  8751                           CL     =$R1
002945  0C4A  8752                           CL     =$R2
002946                                *
002947  0C4B  0000                    CBT40  RESV   1,0                   SPLASH.  ERROR IF O.K. NOT PRINTED
002948                                *                                   IN 5 SEC
002949                                *
002950                                *
002951  0C4C  B840 083F               CBT45  LDR    $R3,STOR1
002952  0C4E  3D01                           CMV    $R3,=1                ZERO SET, BRANCH IF TRUE TESTED?
002953  0C4F  0209                           BL     >CBT50                NO
002954  0C50  3D02                           CMV    $R3,=2                ZERO RESET, BRANCH IF TRUE TESTED?
002955  0C51  0201 001D                      BL     CBT70                 NO
002956  0C53  3D03                           CMV    $R3,=3                ZERO RESET, BRANCH IF FALSE TESTED?
002957  0C54  0201 002B                      BL     CBT80                 NO
002958  0C56  0F81 0040                      B      CBT100                YES
002959                                *
002960                                *
002961                                *
002962  0C58  1D01                    CBT50  CMV    $R1,=1                ERROR?
002963  0C59  0981 0058                      BNE    CBT200                YES.  REPORT ERROR
002964                                *
002965                                *
002966  0C5B  9840 083F                      LDR    $R1,SPHADR            MODIFY MICROCODE FOR TESTING CONDITIONAL
002967  0C5D  1E01                           ADV    $R1,=1                BRANCH WHEN TC = TRUE
002968  0C5E  8051                           IO     =$R1,<OUTADR
        0C5F  0000 14A2
002969  0C61  89C0 0834                      CMZ    RAM2FL                RAM2 PRESENT?
002970  0C63  0981 0046                      BNE    CBT160                YES
002971  0C65  ABC0 FF40                      LAB    $B2,MC5               NO.  FETCH MICROCODE BRANCH TO 0BFB IF
002972  0C67  E3C0 FF9A               CBT60  LNJ    $B6,CSB10             TRUE FROM MAIN MEMORY AND PUT IT INTO
002973  0C69  E3C0 FFA1                      LNJ    $B6,CSB20             WCS
002974  0C6B  8AC0 0820                      INC    STOR1                 BUMP COUNTER
002975  0C6D  0F81 FFD5                      B      CBT30
002976                                *
002977                                *
002978  0C6F  1D00                    CBT70  CMV    $R1,=0                ERROR?
002979  0C70  0981 0041                      BNE    CBT200                YES.  REPORT ERROR
002980                                *
002981                                *
002982                                *
002983                                *
002984                                *  NOW THE TEST CONDITION IS ZERO FLOP RESET (CLEARED)
002985                                *  MODIFY MICROCODE FOR TESTING
002986                                *
002987                                *
002988  0C72  8000 149B                      IO     <SPHADR,<OUTADR       1ST SPLASH LOCATION
        0C74  0000 14A2
002989  0C76  ABC0 FF23                      LAB    $B2,MC2               FETCH MICROCODE RESET ZERO FROM MAIN
002990  0C78  E3C0 FF89                      LNJ    $B6,CSB10             MEMORY AND PUT IT INTO WCS
002991  0C7A  E3C0 FF90                      LNJ    $B6,CSB20
002992  0C7C  8AC0 080F                      INC    STOR1                 BUMP COUNTER
002993  0C7E  0F81 FFC4                      B      CBT30
002994                                *
002995                                *
002996                                *
002997  0C80  1D01                    CBT80  CMV    $R1,=1                ERROR?
002998  0C81  0981 0030                      BNE    CBT200                YES.  REPORT ERROR
002999                                *
003000                                *
003001  0C83  9840 0817                      LDR    $R1,SPHADR            MODIFY MICROCODE FOR TESTING CONDITIONAL
003002  0C85  1E01                           ADV    $R1,=1                BRANCH WHEN TC = FALSE
003003  0C86  8051                           IO     =$R1,<OUTADR
        0C87  0000 14A2
003004  0C89  89C0 080C                      CMZ    RAM2FL                RAM2 PRESENT?
003005  0C8B  0981 0022                      BNE    CBT170                YES
003006  0C8D  ABC0 FF10                      LAB    $B2,MC3               NO.  FETCH MICROCODE BRANCH TO 0BFB IF
003007  0C8F  E3C0 FF72               CBT90  LNJ    $B6,CSB10             FALSE FROM MAIN MEMORY AND PUT IT INTO
003008  0C91  E3C0 FF79                      LNJ    $B6,CSB20             WCS
003009  0C93  8AC0 07F8                      INC    STOR1                 BUMP COUNTER
003010  0C95  0F81 FFAD                      B      CBT30
003011                                *
```

```
003012                                  *
003013                                  *
003014  0C97  1D00                      CBT100 CMV   $R1,=0             ERROR?
003015  0C98  0981 0019                       BNE   CBT200              YES.  REPORT ERROR
003016  0C9A  0F81 0027                       B     CST10               NO.  GO TO NEXT TEST
003017                                  *
003018                                  *
003019  0C9C  ABC0 FF05                  CBT150 LAB   $B2,MC4           FETCH MICROCODE BRANCH TO OFFB IF
003020  0C9E  E3C0 FF63                       LNJ   $B6,CSB10           FALSE FROM MAIN MEMORY AND PUT IT INTO
003021  0CA0  E3C0 FF6A                       LNJ   $B6,CSB20           WCS
003022                                  *
003023  0CA2  ABC0 FF0F                       LAB   $B2,MC8             FETCH MICROCODE INC R1 AND GO TO OFFC
003024  0CA4  E3C0 FF5D                       LNJ   $B6,CSB10           (RETURN ROUTINE)
003025  0CA6  E3C0 FF64                       LNJ   $B6,CSB20
003026  0CA8  0F81 FFBC                       B     CBT20
003027                                  *
003028  0CAA  ABC0 FEFF                  CBT160 LAB   $B2,MC6           FETCH MICROCODE BRANCH TO OFFB IF TRUE
003029  0CAC  0F81 FFBA                       B     CBT60               FROM MAIN MEMORY
003030                                  *
003031  0CAE  ABC0 FEF3                  CBT170 LAB   $B2,MC4           FETCH MICROCODE BRANCH TO OFFB IF
003032  0CB0  0F81 FFDE                       B     CBT90               FALSE FROM MAIN MEMORY
003033                                  *
003034                                  *
003035                                  CBT200 CALL  ZV$ER,CB1,MSGCB1   ERROR "CB1"
        0CB2  FBC0 0003
        0CB4  D380 0000       X
        0CB6  0F80
        0CB7  0CBF
        0CB8  1475
003036                                         CALL  ZV$T,MSG4          PRINT "  * REPLACE MB *"
        0CB9  FBC0 0003
        0CBB  D380 0000       X
        0CBD  0F80
        0CBE  1319
003037  0CBF  0F7F                       CB1   NOP   >$-1                ** ERROR **
003038  0CC0  0F81 FF8B                       B     CBT45               CONTINUE TEST
003039                                  *
003040                                  *
003041                                  *
003042                                  *
003043                                  *
003044                                  *
003045                                  *
003046                                  *  TEST 5
003047                                  *
003048                                  *
003049                                  *  CONDITIONAL SUBROUTINE CALL TEST
003050                                  **********************************
003051                                  *
003052                                  *
003053                                  *  THE CONDITION NOW IS THE ZERO FLOP SET
003054                                  *  MODIFY MICROCODES FOR TESTING
003055                                  *
003056  0CC2  0F7F                       CST10 NOP   >$-1
003057                                  *
003058                                  *
003059  0CC3  8000 149B                         IO    <SPHADR,<OUTADR     WCS 1ST SPLASH LOCATION
        0CC5  0000 14A2
003060  0CC7  8AC0 07D8                         INC   TSTCNT              BUMP TEST COUNTER
003061  0CC9  ABC0 FECC                         LAB   $B2,MC1             FETCH MICROCODE SET ZERO FROM MAIN
003062  0CCB  E3C0 FF36                         LNJ   $B6,CSB10           MEMORY AND PUT IT INTO WCS
003063  0CCD  E3C0 FF3D                         LNJ   $B6,CSB20
003064                                  *
003065  0CCF  89C0 07C6                         CMZ   RAM2FL              RAM2 PRESENT?
003066  0CD1  0981 0077                         BNE   CST150              YES
003067  0CD3  ABC0 FEE6                         LAB   $B2,MC10            NO.  FETCH MICROCODE CALL SUBROUTINE
003068  0CD5  E3C0 FF2C                         LNJ   $B6,CSB10           (AT 0BFB) IF FALSE AND PUT IT INTO WCS
003069  0CD7  E3C0 FF33                         LNJ   $B6,CSB20
003070                                  *
003071  0CD9  B840 07C4                  CST20  LDR   $R3,WCSIZE          LAST LOCATION IN WCS
003072  0CDB  3EFC                             ADV   $R3,=-4             LOCATION FOR SUBROUTINE
003073  0CDC  8053                             IO    =$R3,<OUTADR
        0CDD  0000 14A2
003074  0CDF  ABC0 FEEA                         LAB   $B2,MC14            FETCH SUBROUTINE (INC R2 AND RETURN)
003075  0CE1  E3C0 FF20                         LNJ   $B6,CSB10           AND PUT IT INTO WCS
003076  0CE3  E3C0 FF27                         LNJ   $B6,CSB20
003077                                  *
003078                                  *
003079                                  *  STARTS TEST HERE
003080                                  *
003081  0CE5  8740 07A6                         CL    STOR1               COUNTER FOR CHECKING ALL TEST
003082                                  *                                 CONDITIONS ARE TESTED
003083  0CE7  9840 07B4                  CST30  LDR   $R1,SPHCOD          SPLASH MICROCODE
003084  0CE9  9F40 0005                         STR   $R1,CS140
003085                                  *
003086                                  *
003087  0CEB  E3C0 FDAB                         LNJ   $B6,SRT210          ENABLE WCS AND STATUS CHECK
003088                                  *
003089  0CED  8751                             CL    =$R1
003090  0CEE  8752                             CL    =$R2
003091                                  *
003092  0CEF  0000                       CST40  RESV  1,0                 SPLASH.  ERROR IF O.K. NOT PRINTED
003093                                  *                                 IN 5 SEC
003094                                  *
003095                                  *
003096                                  *
003097  0CF0  B840 079B                  CST45  LDR   $R3,STOR1
003098  0CF2  3D01                             CMV   $R3,=1              ZERO SET, CALL IF TRUE TESTED?
003099  0CF3  0209                             BL    >CST50              NO
003100  0CF4  3D02                             CMV   $R3,=2              ZERO RESET, CALL IF TRUE TESTED?
003101  0CF5  0201 0020                        BL    CST70               NO
003102  0CF7  3D03                             CMV   $R3,=3              ZERO RESET, CALL IF FALSE TESTED?
003103  0CF8  0201 0031                        BL    CST80               NO
003104  0CFA  0F81 0046                        B     CST100              YES
003105                                  *
003106                                  *
003107  0CFC  1D01                       CST50  CMV   $R1,=1              ERROR?
003108  0CFD  0981 005B                        BNE   CST200              YES.  REPORT ERROR
003109  0CFF  2D00                             CMV   $R2,=0              ERROR?
003110  0D00  0981 0058                        BNE   CST200              YES.  REPORT ERROR
003111                                  *
003112                                  *
003113  0D02  9840 0798                         LDR   $R1,SPHADR          MODIFY MICROCODE FOR TESTING CONDITIONAL
```

```
003114  0D04  1E01                    ADV   $R1,=1              SUBROUTINE CALL WHEN TC= TRUE
003115  0D05  8051                    IO    =$R1,<OUTADR
        0D06  0000 14A2
003116  0D08  89C0 078D                CMZ   RAM2FL              RAM2 PRESENT?
003117  0D0A  0981 0046                BNE   CST160              YES
003118  0D0C  ABC0 FEB5                LAB   $B2,MC12            NO.  FETCH MICROCODE CALL SUBROUTINE
003119  0D0E  E3C0 FEF3      CST60     LNJ   $B6,CSB10           (AT 0BFB) IF TRUE AND PUT IT INTO WCS
003120  0D10  E3C0 FEFA                LNJ   $B6,CSB20
003121  0D12  8AC0 0779-               INC   STOR1               BUMP COUNTER
003122  0D14  0F81 FFD2                B     CST30
003123                                *
003124                                *
003125  0D16  1D01            CST70     CMV   $R1,=1              ERROR?
003126  0D17  0981 0041                BNE   CST200              YES.  REPORT ERROR
003127  0D19  2D01                     CMV   $R2,=1              ERROR?
003128  0D1A  0981 003E                BNE   CST200              YES.  REPORT ERROR
003129                                *
003130                                *
003131                                *  NOW THE TEST CONDITION IS ZERO FLOP CLEARED
003132                                *  MODIFY MICROCODE FOR TESTING
003133                                *
003134                                *
003135  0D1C  8000 149B                IO    <SPHADR,<OUTADR     WCS 1ST SPLASH LOCATION
        0D1E  0000 14A2
003136  0D20  ABC0 FE79                LAB   $B2,MC2             FETCH MICROCODE RESET ZERO FROM MAIN
003137  0D22  E3C0 FEDF                LNJ   $B6,CSB10           MEMORY AND PUT IT INTO WCS
003138  0D24  E3C0 FEE6                LNJ   $B6,CSB20
003139  0D26  8AC0 0765                INC   STOR1               BUMP COUNTER
003140  0D28  0F81 FFBE                B     CST30
003141                                *
003142  0D2A  1D01            CST80     CMV   $R1,=1              ERROR?
003143  0D2B  0981 002D                BNE   CST200              YES.  REPORT ERROR
003144  0D2D  2D00                     CMV   $R2,=0              ERROR?
003145  0D2E  0981 002A                BNE   CST200              YES.  REPORT ERROR
003146                                *
003147  0D30  9840 076A                LDR   $R1,SPHADR          MODIFY MICROCODE FOR TESTING CONDITIONAL
003148  0D32  1E01                     ADV   $R1,=1              SUBROUTINE CALL WHEN TC = FALSE
003149  0D33  89C0 0762                CMZ   RAM2FL              RAM2 PRESENT?
003150  0D35  0981 001F                BNE   CST170              YES
003151  0D37  ABC0 FE82                LAB   $B2,MC10            NO.  FETCH MICROCODE CALL SUBROUTINE
003152  0D39  E3C0 FEC8      CST90     LNJ   $B6,CSB10           (AT 0BFB) IF FALSE AND PUT  IT INTO WCS
003153  0D3B  E3C0 FECF                LNJ   $B6,CSB20
003154  0D3D  8AC0 074E                INC   STOR1               BUMP COUNTER
003155  0D3F  0F81 FFA7                B     CST30
003156                                *
003157                                *
003158  0D41  1D01            CST100    CMV   $R1,=1              ERROR?
003159  0D42  0981 0016                BNE   CST200              YES.  REPORT ERROR
003160  0D44  2D01                     CMV   $R2,=1              ERROR?
003161  0D45  0981 0013                BNE   CST200              YES.  REPORT ERROR
003162  0D47  0F81 0021                B     CRT10               NO.  GO TO NEXT TEST
003163                                *
003164                                *
003165                                *
003166  0D49  ABC0 FE74      CST150    LAB   $B2,MC11            FETCH MICROCODE CALL SUBROUTINE (AT
003167  0D4B  E3C0 FEB6                LNJ   $B6,CSB10           0FFB) IF FALSE AND PUT IT INTO WCS
003168  0D4D  E3C0 FEBD                LNJ   $B6,CSB20
003169  0D4F  0F81 FF89                B     CST20
003170                                *
003171  0D51  ABC0 FE74      CST160    LAB   $B2,MC13            FETCH MICROCODE CALL SUBROUTINE (AT
003172  0D53  0F81 FFBA                B     CST60               0FFB) IF TRUE FROM MAIN MEMORY
003173                                *
003174  0D55  ABC0 FE68      CST170    LAB   $B2,MC11            FETCH MICROCODE CALL SUBROUTINE (AT
003175  0D57  0F81 FFE1                B     CST90               0FFB) IF FALSE FROM MAIN MEMORY
003176                                *
003177                                *
003178                                *
003179                                *
003180                        CST200  CALL  ZV$ER,CS1,MSGCS1    ERROR "CS1"
        0D59  FBC0 0003
        0D5B  D380 0000         X
        0D5D  0F80
        0D5E  0D66
        0D5F  1477
003181                                CALL  ZV$T,MSG4           PRINT "  * REPLACE MB *"
        0D60  FBC0 0003
        0D62  D380 0000         X
        0D64  0F80
        0D65  1319
003182  0D66  0F7F            CS1       NOP   >$-1                **  ERROR  **
003183  0D67  0F81 FF88                B     CST45               CONTINUE TEST
003184                                *
003185                                *
003186                                *
003187                                *
003188                                *  TEST 6
003189                                *
003190                                *
003191                                *  CONDITIONAL RETURN TEST
003192                                ************************
003193                                *
003194                                *
003195                                *  THE TEST CONDITION IS ZERO FLOP SET
003196                                *
003197                                *
003198  0D69  0F7F            CRT10     NOP   >$-1
003199                                *
003200                                *
003201  0D6A  8000 149B                IO    <SPHADR,<OUTADR     WCS 1ST SPLASH LOCATION
        0D6C  0000 14A2
003202  0D6E  8AC0 0731                INC   TSTCNT              BUMP TEST COUNTER
003203  0D70  ABC0 FE25                LAB   $B2,MC1             FETCH MICROCODE SET ZERO FROM MAIN
003204  0D72  E3C0 FE8F                LNJ   $B6,CSB10           MEMORY AND PUT IT INTO WCS
003205  0D74  E3C0 FE96                LNJ   $B6,CSB20
003206                                *
003207  0D76  89C0 071F                CMZ   RAM2FL              RAM2  PRESENT?
003208  0D78  0981 0072                BNE   CRT150              YES
003209  0D7A  ABC0 FE53                LAB   $B2,MC15            NO.  FETCH MICROCODE UNCONDITIONAL
003210  0D7C  E3C0 FE85                LNJ   $B6,CSB10           CALL FROM MAIN MEMORY AND PUT IT INTO
003211  0D7E  E3C0 FE8C                LNJ   $B6,CSB20           WCS
003212                                *
003213  0D80  B840 071D      CRT20     LDR   $R3,wCSIZE          LAST LOCATION IN WCS
003214  0D82  3EFC                     ADV   $R3,=-4             LOCATION FOR CONDITIONAL RETURN
```

```
003215  0D83  8053                    IO    =$R3,<OUTADR
        0D84  0000 14A2
003216  0D86  ABC0 FE57               LAB   $B2,MC19          FETCH MICROCODE RETURN IF FALSE
003217  0D88  89C0 070D               CMZ   RAM2FL            RAM2 PRESENT?
003218  0D8A  0981 0068               BNE   CRT160            YES
003219  0D8C  E3C0 FE75       CRT25   LNJ   $B6,CSB10         NO.  LOAD WCS
003220  0D8E  E3C0 FE7C               LNJ   $B6,CSB20
003221                        *
003222                        *  STARTS TEST HERR
003223                        *
003224  0D90  8740 06FB               CL    STOR1             COUNTER FOR CHECKING ALL TEST
003225                        *                               CONDITIONS ARE TESTED
003226  0D92  9840 0709       CRT30   LDR   $R1,SPHCOD        SPLASH MICROCODE
003227  0D94  9F40 0005               STR   $R1,CRT40
003228                        *
003229                        *
003230  0D96  E3C0 FD00               LNJ   $B6,SRT210        ENABLE WCS AND STATUS  CHECK
003231                        *
003232  0D98  8751                    CL    =$R1
003233  0D99  8752                    CL    =$R2
003234
003235  0D9A  0000            CRT40   RESV  1,0               SPLASH.  ERROR IF O.K. NOT PRINTED
003236                        *                               IN 5 SEC
003237                        *
003238  0D9B  B840 06F0       CRT45   LDR   $R3,STOR1
003239  0D9D  3D01                    CMV   $R3,=1            ZERO SET, RETURN IF TRUE TESTED?
003240  0D9E  0209                    BL    >CRT50            NO
003241  0D9F  3D02                    CMV   $R3,=2            ZERO RESET, RETURN IF TRUE TESTED?
003242  0DA0  0201 001D               BL    CRT70             NO
003243  0DA2  3D03                    CMV   $R3,=3            ZERO RESET, RETURN IF FALSE TESTED?
003244  0DA3  0201 002B               BL    CRT80             NO
003245  0DA5  0F81 0040               B     CRT100            YES
003246                        *
003247  0DA7  1D00            CRT50   CMV   $R1,=0            ERROR?
003248  0DA8  0981 0056               BNE   CRT200            YE.  REPORT ERROR
003249                        *
003250  0DAA  B840 06F3               LDR   $R3,WCSIZE        MODIFY MICROCODE FOR TESTING CONDITIONAL
003251  0DAC  3EFC                    ADV   $R3,=-4           RETURN WHEN TC = TRUE
003252  0DAD  8053                    IO    =$R3,<OUTADR
        0DAE  0000 14A2
003253  0DB0  ABC0 FE25               LAB   $B2,MC17          FETCH MICROCODE RETURN IF TRUE
003254  0DB2  89C0 06E3               CMZ   RAM2FL            RAM2 PRESENT?
003255  0DB4  0981 0042               BNE   CRT170            YES
003256  0DB6  E3C0 FE4B       CRT60   LNJ   $B6,CSB10         NO.  LOAD WCS
003257  0DB8  E3C0 FE52               LNJ   $B6,CSB20
003258  0DBA  8AC0 06D1               INC   STOR1             BUMP COUNTER
003259  0DBC  0F81 FFD5               B     CRT30
003260                        *
003261                        *
003262  0DBE  1D01            CRT70   CMV   $R1,=1            ERROR?
003263  0DBF  0981 003F               BNE   CRT200            YES.  REPORT ERROR
003264                        *
003265                        *
003266                        *  NOW THE TEST CONDITION IS ZERO FLOP RESET (CLEARED)
003267                        *  MODIFY MICROCODE FOR TESTING
003268                        *
003269  0DC1  8000 149B               IO    <SPHADR,<OUTADR   WCS 1ST SPLASH LOCATION
        0DC3  0000 14A2
003270  0DC5  ABC0 FDD4               LAB   $B2,MC2           FETCH MICROCODE RESET ZERO FROM MAIN
003271  0DC7  E3C0 FE3A               LNJ   $B6,CSB10         MEMORY AND PUT IT INTO WCS
003272  0DC9  E3C0 FE41               LNJ   $B6,CSB20
003273  0DCB  8AC0 06C0               INC   STOR1             BUMP COUNTER
003274  0DCD  0F81 FFC4               B     CRT30
003275                        *
003276  0DCF  1D00            CRT80   CMV   $R1,=0            ERROR?
003277  0DD0  0981 002E               BNE   CRT200            YES.  REPORT ERROR
003278                        *
003279  0DD2  B840 06CB               LDR   $R3,WCSIZE        LAST LOCATION IN WCS
003280  0DD4  3EFC                    ADV   $R3,=-4           LOCATION FOR CONDITIONAL RETURN
003281  0DD5  8053                    IO    =$R3,<OUTADR
        0DD6  0000 14A2
003282  0DD8  ABC0 FE05               LAB   $B2,MC19          FETCH MICROCODE RETURN IF FALSE
003283  0DDA  89C0 06BB               CMZ   RAM2FL            RAM2 PRESENT?
003284  0DDC  0981 001E               BNE   CRT180            YES
003285  0DDE  E3C0 FE23       CRT90   LNJ   $B6,CSB10         NO.  LOAD WCS
003286  0DE0  E3C0 FE2A               LNJ   $B6,CSB20
003287  0DE2  8AC0 06A9               INC   STOR1             BUMP COUNTER
003288  0DE4  0F81 FFAD               B     CRT30
003289                        *
003290                        *
003291  0DE6  1D01            CRT100  CMV   $R1,=1            ERROR?
003292  0DE7  0981 0017               BNE   CRT200            YES.  REPORT ERROR
003293  0DE9  0F81 0025               B     LBT10             NO.  GO TO NEXT TEST
003294                        *
003295                        *
003296  0DEB  ABC0 FDE6       CRT150  LAB   $B2,MC16          FETCH MICROCODE UNCONDITIONAL CALL
003297  0DED  E3C0 FE14               LNJ   $B6,CSB10         FROM MAIN MEMORY AND PUT IT INTO WCS
003298  0DEF  E3C0 FE1B               LNJ   $B6,CSB20
003299  0DF1  0F81 FF8E               B     CRT20
003300                        *
003301  0DF3  ABC0 FDEE       CRT160  LAB   $B2,MC20          FETCH MICROCODE RETURN IF FALSE FROM
003302  0DF5  0F81 FF96               B     CRT25             MAIN MEMORY
003303                        *
003304  0DF7  ABC0 FDE2       CRT170  LAB   $B2,MC18          FETCH MICROCODE RETURN IF TRUE FROM
003305  0DF9  0F81 FFBC               B     CRT60             MAIN MEMORY
003306                        *
003307  0DFB  ABC0 FDE6       CRT180  LAB   $B2,MC20          FETCH MICROCODE RETURN IF FALSE FROM
003308  0DFD  0F81 FFE0               B     CRT90             MAIN MEMORY
003309                        *
003310                        *
003311  0DFF  FBC0 0003       CRT200  CALL  ZV$ER,CR1,MSGCR1  ERROR "CR1"
        0E01  D380 0000       X
        0E03  0F80
        0E04  0E0C
        0E05  1479
003312  0E06  FBC0 0003               CALL  ZV$T,MSG4         PRINT "  * REPLACE MB *"
        0E08  D380 0000       X
        0E0A  0F80
        0E0B  1319
003313  0E0C  0F7F            CR1     NOP   >$-1              **  ERROR  **
003314  0E0D  0F81 FF8D               B     CRT45
```

```
003315                              *
003316                              *
003317                              *
003318                              *
003319                              *
003320                              *  TEST 7
003321                              *
003322                              *
003323                              *  L - BRANCH TEST
003324                              ******************
003325                              *
003326                              *
003327                              *  PRELOAD MICROCODES IN WCS
003328                              *
003329  0E0F  0F7F                  LBT10  NOP   >$-1
003330                              *
003331                              *
003332  0E10  8040 068A                    IO    SPHADR,<OUTADR        WCS 1ST LOCATION
        0E12  0000 14A2
003333  0E14  8AC0 068B                    INC   TSTCNT                BUMP TEST COUNTER
003334  0E16  ABC0 FDD7                    LAB   $B2,MC23              FETCH MICROCODE INC R1 FROM MAIN
003335  0E18  E3C0 FDE9                    LNJ   $B6,CSB10             MEMORY AND PUT IT INTO WCS
003336  0E1A  E3C0 FDF0                    LNJ   $B6,CSB20
003337                              *
003338  0E1C  ABC0 FDC9                    LAB   $B2,MC21              FETCH MICROCODE INC R1 AND PUT RESULT
003339  0E1E  E3C0 FDE3                    LNJ   $B6,CSB10             IN LINK REGISTER
003340  0E20  E3C0 FDEA                    LNJ   $B6,CSB20             LOAD MICROCODE IN WCS
003341                              *
003342  0E22  ABC0 FDC7                    LAB   $B2,MC22              FETCH MICROCODE L - BRANCH FROM MAIN
003343  0E24  E3C0 FDDD                    LNJ   $B6,CSB10             MEMORY AND LOAD WCS
003344  0E26  E3C0 FDE4                    LNJ   $B6,CSB20
003345                              *
003346  0E28  A840 0672                    LDR   $R2,SPHADR
003347  0E2A  2E03                         ADV   $R2,=3                STARTING LOCATION TO FILL WCS WITH
003348  0E2B  AF40 0660                    STR   $R2,STOR1             INC R2.  SAVE STARTING LOCATION
003349  0E2D  B840 0670                    LDR   $R3,WCSIZE            LAST LOCATION IN WCS
003350  0E2F  3EFC                         ADV   $R3,=-4               ENDING LOCATION FOR INC R2
003351                              *
003352  0E30  ABC0 FD85                    LAB   $B2,MC9               FETCH MICROCODE INC R2
003353  0E32  E3C0 FDCF                    LNJ   $B6,CSB10
003354  0E34  E3C0 FDD6            LBT20   LNJ   $B6,CSB20             LOAD WCS
003355  0E36  2E01                         ADV   $R2,=1
003356  0E37  A953                         CMR   $R2,=$R3              ALL LOCATIONS FILLED?
003357  0E38  03FC                         BLE   >LBT20                NO
003358                              *
003359  0E39  0F83                         B     >LBT25
003360  0E3A  E3C0 03FC                    LNJ   $B6,RDBK10            JUMP TO READ BACK SUBROUTINE
003361                              *
003362                              *
003363                              *  STARTS TEST HERE
003364                              *
003365  0E3C  9840 065F            LBT25   LDR   $R1,SPHCOD            SPLASH MICROCODE
003366  0E3E  9F40 000D                    STR   $R1,LBT40
003367                              *
003368                              *
003369  0E40  BBC0 002B                    LAB   $B3,LBT48
003370                              *
003371  0E42  B840 065B                    LDR   $R3,WCSIZE
003372  0E44  3EFD                         ADV   $R3,=-3
003373  0E45  5C03                         LDV   $R5,=3                TO CHECK L - REGISTER WRAP AROUND
003374  0E46  1C04                         LDV   $R1,=4                INITIALIZE R1
003375  0E47  C840 0644                    LDR   $R4,STOR1             STARTING LOCATION FOR INC R2
003376  0E49  E3C0 FC4D            LBT30   LNJ   $B6,SRT210            ENABLE WCS AND STATUS CHECK
003377  0E4B  A854                         LDR   $R2,=$R4              INITIALIZE R2
003378                              *
003379  0E4C  0000                 LBT40   RESV  1,0                   SPLASH.  ERROR IF O.K. NOT PRINTED
003380                              *                                   IN 5 SEC
003381                              *
003382  0E4D  A953                         CMR   $R2,=$R3              ERROR?
003383  0E4E  09A6                         BNE   >LBT50                YES.  REPORT ERROR
003384  0E4F  5E01                         ADV   $R5,=1                COUNT 1
003385  0E50  D970 00FF                    CMR   $R5,=Z'00FF'          L - REGISTER WRAP AROUND?
003386  0E52  0306                         BG    >LBT45                YES
003387                              *
003388                              *
003389  0E53  4E01                 LBT43   ADV   $R4,=1                NEXT LOCATION FOR L - BRANCH TESTING
003390  0E54  C953                         CMR   $R4,=$R3              L - BRANCH TESTED FOR ALL LOCATIONS?
003391  0E55  0274                         BL    >LBT30                NO
003392  0E56  0F81 00F5                    B     IFT10                 YES.  GO  TO NEXT TEST
003393                              *
003394                              *
003395  0E58  E840 0642            LBT45   LDR   $R6,SPHADR
003396  0E5A  6E02                         ADV   $R6,=2
003397  0E5B  8056                         IO    =$R6,<OUTADR
        0E5C  0000 14A2
003398  0E5E  CF40 0629                    STR   $R4,TMP2              SAVE R4
003399                              *
003400  0E60  ABC0 FD89                    LAB   $B2,MC22
003401  0E62  E3C0 FD9F                    LNJ   $B6,CSB10
003402  0E64  FA73                         ADD   $R7,+$B3
003403  0E65  E3C0 FDA5                    LNJ   $B6,CSB20
003404  0E67  C840 0620                    LDR   $R4,TMP2              RESTORE R4
003405                              *
003406  0E69  1CFE                         LDV   $R1,=-2               REINITIALIZE R1
003407  0E6A  5C00                         LDV   $R5,=0                REINITIALIZE R5
003408  0E6B  0FE8                         B     >LBT43
003409                              *
003410                              *
003411  0E6C  0100                 LBT48   DC    X'100'
003412  0E6D  0200                         DC    X'200'
003413  0E6E  0300                         DC    X'300'
003414  0E6F  0400                         DC    X'400'
003415  0E70  0500                         DC    X'500'
003416  0E71  0600                         DC    X'600'
003417  0E72  0700                         DC    X'700'
003418  0E73  0800                         DC    X'800'
003419                              *
003420                              *
003421                              *
003422                              LBT50   CALL  ZV$ER,LB1,MSGLB1      ERROR "LB1"
        0E74  FBC0 0003
        0E76  0380 0000       X
        0E78  0F80
```

```
                0E79  0E81
                0E7A  147B
003423                                  CALL  ZV$T,MSG4           PRINT "  * REPLACE MB *"
                0E7B  FBC0 0003
                0E7D  D380 0000     X
                0E7F  0F80
                0E80  1319
003424          0E81  0F7F              LB1   NOP   >$-1            **  ERROR  **
003425          0E82  0F81 0001              B     ZNT10           GO TO NEXT TEST
003426                                  *
003427                                  *
003428                                  *
003429                                  *
003430                                  *
003431                                  *  TEST 8
003432                                  *
003433                                  *
003434                                  *  ZERO NA FIELD TEST
003435                                  ********************
003436                                  *
003437                                  *
003438          0E84  0F7F              ZNT10 NOP   >$-1
003439                                  *
003440                                  *
003441          0E85  8000 149B               IO    <SPHADR,<OUTADR   WCS 1ST ENTRY LOCATION
                0E87  0000 14A2
003442          0E89  8AC0 0616               INC   TSTCNT            BUMP TEST COUNTER
003443          0E8B  9840 060F               LDR   $R1,SPHADR        SPLASH ADDR.  R1 = COUNTER
003444          0E8D  B840 0610               LDR   $R3,WCSIZE        LAST LOCATION IN  WCS
003445          0E8F  3EFC                    ADV   $R3,=-4           LAST LOCATION FOR TESTING
003446          0E90  ABC0 FD5D               LAB   $B2,MC23          FETCH MICROCODE FOR INC R1
003447          0E92  E3C0 FD6F               LNJ   $B6,CSB10
003448          0E94  8757                    CL    =$R7              CHANGE NA FIELD TO  0
003449          0E95  E3C0 FD75         ZNT20 LNJ   $B6,CSB20         LOAD WCS
003450          0E97  1E01                    ADV   $R1,=1
003451          0E98  9953                    CMR   $R1,=$R3          ALL LOCATIONS LOADED?
003452          0E99  03FC                    BLE   >ZNT20            NO
003453                                  *
003454                                  *
003455          0E9A  9870 0140               LDR   $R1,=Z'0140'
003456          0E9C  9900 148E               CMR   $R1,<STABIT       TRANSPARENT MODE?
003457          0E9E  0985                    BNE   >ZNT25            NO
003458          0E9F  7C02                    LDV   $R7,=2            YES.  MODIFY NA FIELD
003459          0EA0  B270 0800               SUB   $R3,=Z'0800'
003460          0EA2  0F87                    B     >ZNT30
003461                                  *
003462                                  *
003463          0EA3  B470 8000         ZNT25 OR    $R3,=Z'8000'
003464          0EA5  B270 0800               SUB   $R3,=Z'0800'
003465          0EA7  F870 8002               LDR   $R7,=Z'8002'      NA FIELD
003466          0EA9  8000 149B         ZNT30 IO    <SPHADR,<OUTADR   WCS 1ST ENTRY LOCATION
                0EAB  0000 14A2
003467          0EAD  C870 0093               LDR   $R4,=Z'0093'      MICROCODE UNCONDITIONAL JUMP
003468          0EAF  D870 C700               LDR   $R5,=Z'C700'
003469          0EB1  E870 2000               LDR   $R6,=Z'2000'
003470                                  *
003471          0EB3  E3C0 FD57               LNJ   $B6,CSB20         LOAD WCS
003472          0EB5  0F83                    B     >ZNT35            CHANGE TO  NOP MANUALLY FOR  READ BACK
003473          0EB6  E3C0 0380               LNJ   $B6,RDBK10        JUMP TO READ BACK SUBROUTINE
003474                                  *
003475                                  *  STARTS TEST HERE
003476                                  *
003477          0EB8  AB80 0ECF         ZNT35 LAB   $B2,<ZNT80        SET UP IV 15
003478          0EBA  AF80 0000     X         STB   $B2,<ZHIH15
003479          0EBC  A840 05DF               LDR   $R2,SPHCOD        SPLASH MICROCODE
003480          0EBE  AF40 0003               STR   $R2,ZNT40
003481                                  *
003482                                  *
003483          0EC0  E3C0 FBD6               LNJ   $B6,SRT210        ENABLE WCS AND STATUS CHECK
003484                                  *
003485                                  *
003486                                  *
003487          0EC2  0000              ZNT40 RESV  1,0               SPLASH.  ERROR IF O.K. NOT  PRINTED
003488                                  *                             IN 5 SEC
003489          0EC3  89C0 05DB               CMZ   TRPFLG            TRAP?
003490          0EC5  0901 000C               BE    ZNT90             NO.  REPORT ERROR
003491          0EC7  8740 05D7               CL    TRPFLG            YES.  CLEAR TRAP FLAG
003492                                  *
003493          0EC9  7E01              ZNT50 ADV   $R7,=1            BUMP NA FIELD FOR MICROCODE UNCONDITIONAL
003494                                  *                             JUMP
003495          0ECA  F953                    CMR   $R7,=$R3          ALL LOCATIONS TESTED?
003496          0ECB  0381 FFDD               BLE   ZNT30             NO
003497          0ECD  0F81 0014               B     SAT10             YES.  GO TO NEXT TEST
003498                                  *
003499                                  *
003500                                  *
003501          0ECF  8AC0 05CF         ZNT80 INC   TRPFLG            SET TRAP FLAG
003502          0ED1  0003                    RTT                     RETURN
003503                                  *
003504                                  *
003505                                  ZNT90 CALL  ZV$ER,ZN1,MSGZN1  ERROR "ZN1"
                0ED2  FBC0 0003
                0ED4  D380 0000     X
                0ED6  0F80
                0ED7  0EDF
                0ED8  147D
003506                                        CALL  ZV$T,MSG4         PRINT "*  REPLACE MB*"
                0ED9  FBC0 0003
                0EDB  D380 0000     X
                0EDD  0F80
                0EDE  1319
003507          0EDF  0F7F              ZN1   NOP   >$-1              **  ERROR  **
003508                                  *                             ZERO NA FIELD ERROR
003509                                  *                             REPLACE MOTHER BOARD
003510          0EE0  0F81 0001               B     SAT10             GO TO NEXT TEST
003511                                  *
003512                                  *
003513                                  *
003514                                  *
003515                                  *  TEST 9
003516                                  *
003517                                  *
003518                                  *  SPLASH AFTER REINITIALIZE WCS TEST
```

```
003519                                  ************************************
003520                                  *
003521                                  *
003522  0EE2  0F7F                      SAT10   NOP     >$-1
003523                                  *
003524                                  *
003525  0EE3  8000 149B                         IO      <SPHADR,<OUTADR         WCS 1ST ENTRY LOCATION
        0EE5  0000 14A2
003526                                  *
003527  0EE7  8AC0 05B8                         INC     TSTCNT                  BUMP TEST COUNTER
003528                                  *
003529                                  *
003530  0EE9  AB80 0F25                         LAB     $B2,<SAT90              SET UP FOR TV5
003531  0EEB  AF80 0000         X               STB     $B2,<ZHTH5
003532  0EED  3CF0                              LDV     $R3,=-16                TO COUNT 16 SPLASH LOCATIONS
003533  0EEE  1CF0                              LDV     $R1,=-16                PRELOAD 16 LOCATIONS WITH MICROCODES
003534  0EEF  89C0 05A6                         CMZ     RAM2FL                  RAM2 PRESENT?
003535  0EF1  0981 002F                         BNE     SAT80                   YES
003536  0EF3  ABC0 FCBA                         LAB     $B2,MC7                 NO.  FETCH MICROCODE (INCR D1;D1  RET1)
003537                                  *
003538                                  *
003539  0EF5  A870 0140                 SAT20   LDR     $R2,=Z'0140'
003540  0EF7  A900 148E                         CMR     $R2,<STABIT             TRANSPARENT MODE?
003541  0EF9  0983                              BNE     >SAT25                  NO
003542  0EFA  F270 8000                         SUB     $R7,=Z'8000'            YES.  MODIFY NA FIELD
003543                                  *
003544                                  *
003545  0EFC  E3C0 FD05                 SAT25   LNJ     $B6,CSB10               LOAD WCS
003546  0EFE  E3C0 FD0C                         LNJ     $B6,CSB20
003547  0F00  17FC                              BINC    $R1,>SAT25              BRANCH IF ALL 16 LOC. ARE NOT LOADED
003548  0F01  0F83                              B       >SAT30
003549  0F02  E3C0 0334                         LNJ     $B6,RDBK10              JUMP TO READ BACK SUBROUTINE
003550                                  *
003551                                  *  STARTS TEST HERE
003552                                  *
003553  0F04  A840 0597                 SAT30   LDR     $R2,SPHCOD              SPLASH MICROCODE
003554  0F06  AF40 000B                 SAT32   STR     $R2,SAT40
003555  0F08  E3C0 FB8E                         LNJ     $B6,SRT210              ENABLE WCS AND STATUS CHECK
003556                                  *
003557                                  *
003558                                  *
003559  0F0A  8070 8000                 SAT35   IO      =Z'8000',<OUTCON        SOFT INITIALIZE WCS
        0F0C  0000 14A1
003560  0F0E  0703                              BIOT    >SAT38
003561  0F0F  E380 114B                         LNJ     $B6,<NAKERR             REPORT "NAK" ERROR
003562  0F11  8751                      SAT38   CL      =$R1
003563                                  *
003564                                  *
003565  0F12  0000                      SAT40   RESV    1,0                     SPLASH
003566                                  *
003567  0F13  89C0 058B                         CMZ     TRPFLG                  TRAP?
003568  0F15  0901 0012                         BE      SAT100                  NO.  REPORT ERROR
003569  0F17  8740 0587                         CL      TRPFLG                  YES.  CLEAR TRAP FLAG
003570  0F19  1D00                              CMV     $R1,=0                  ERROR?
003571  0F1A  0981 001D                         BNE     SAT110                  YES.  REPORT ERROR
003572  0F1C  8AD2                              INC     =$R2                    BUMP GENERIC FOR NEXT SPLASH TEST
003573  0F1D  3781 FFE8                         BINC    $R3,SAT32               BRANCH TO CONTINUE NEXT SPLASH TEST
003574  0F1F  0F81 0026                         B       SAT120
003575                                  *
003576                                  *
003577                                  *
003578  0F21  ABC0 FC90                 SAT80   LAB     $B2,MC8                 MICROCODE (INCR D1;D1 RET2)
003579  0F23  0F81 FFD1                         B       SAT20
003580                                  *
003581                                  *
003582  0F25  8AC0 0579                 SAT90   INC     TRPFLG                  SET TRAP FLAG
003583  0F27  0003                              RTT
003584                                  *
003585                                  *
003586                                  SAT100  CALL    ZV$ER,SA1,MSGSA1        ERROR "SA1"
        0F28  FBC0 0003
        0F2A  D380 0000         X
        0F2C  0F80
        0F2D  0F35
        0F2E  1483
003587                                          CALL    ZV$T,MSG4               PRINT "*  REPLACE MB *"
        0F2F  FBC0 0003
        0F31  D380 0000         X
        0F33  0F80
        0F34  1319
003588  0F35  0F7F                      SA1     NOP     >$-1                    **  ERROR  **
003589                                  *                                       REPLACE MOTHER BOARD
003590  0F36  0F81 000F                         B       SAT120
003591                                  *
003592                                  *
003593                                  SAT110  CALL    ZV$ER,SA2,MSGSA2        ERROR "SA2"
        0F38  FBC0 0003
        0F3A  D380 0000         X
        0F3C  0F80
        0F3D  0F45
        0F3E  1485
003594                                          CALL    ZV$T,MSG4               PRINT "*  REPLACE MB *"
        0F3F  FBC0 0003
        0F41  D380 0000         X
        0F43  0F80
        0F44  1319
003595  0F45  0F7F                      SA2     NOP     >$-1
003596                                  *
003597  0F46  A870 0140                 SAT120  LDR     $R2,=Z'0140'
003598  0F48  A900 148E                         CMR     $R2,<STABIT             TRANSPARENT MODE?
003599  0F4A  0901 0196                         BE      TIT390                  YES
003600                                  *
003601                                  *
003602                                  *
003603                                  *
003604                                  *
003605                                  *
003606                                  *
003607                                  *
003608                                  *
003609                                  *  TEST 10
003610                                  *
003611                                  *
```

```
003612                                  *  INTERFACE TEST
003613                                  ****************
003614                                  *
003615                                  *
003616                                  *  THE PURPOSE OF THIS TEST IS TO RUN ADDITIONAL FIRMFARE FOR CPU-WCS
003617                                  *  INTERFACE TEST
003618                                  *
003619  0F4C  0F7F                      IFT10  NOP    >$-1
003620                                  *
003621                                  *
003622  0F4D  8000 149B                        IO     <SPHADR,<OUTADR       WCS 1ST SPLASH LOCATION
      0F4F  0000 14A2
003623  0F51  8AC0 054E                        INC    TSTCNT                BUMP TEST COUNTER
003624                                  *
003625                                  *
003626  0F53  ABC0 FC9E                        LAB    $B2,MC24              FETCH MICROCODE SET SEL REGISTER FROM
003627  0F55  E3C0 FCAC                        LNJ    $B6,CSB10             MAIN MEMORY AND PUT IT INTO WCS
003628  0F57  E3C0 FCB3                        LNJ    $B6,CSB20
003629                                  *
003630  0F59  89C0 053C                        CMZ    RAM2FL                RAM2 PRESENT?
003631  0F5B  0981 0037                        BNE    IFT60                 YES
003632  0F5D  ABC0 FC98                        LAB    $B2,MC25              NO.  FETCH MICROCODE BRANCH TO 0BFB IF
003633  0F5F  E3C0 FCA2                        LNJ    $B6,CSB10             TRUE FOR SEL REG SET FROM MAIN MEMORY
003634  0F61  E3C0 FCA9                        LNJ    $B6,CSB20             AND PUT IT INTO WCS
003635                                  *
003636  0F63  ABC0 FC4A                        LAB    $B2,MC7               FETCH MICROCODE INC R1 AND GO TO 0BFC
003637  0F65  E3C0 FC9C                        LNJ    $B6,CSB10             (RETURN ROUTINE) AND PUT IT INTO WCS
003638  0F67  E3C0 FCA3                        LNJ    $B6,CSB20
003639                                  *
003640  0F69  B840 0534                 IFT20  LDR    $R3,WCSIZE            LAST LOCATION IN WCS
003641  0F6B  3EFC                             ADV    $R3,=-4               LOCATION FOR MICROCODE INC B7
003642  0F6C  8053                             IO     =$R3,<OUTADR
      0F6D  0000 14A2
003643  0F6F  ABC0 FC8E                        LAB    $B2,MC27              FETCH MICROCODE INC B7 FROM MAIN
003644  0F71  E3C0 FC90                        LNJ    $B6,CSB10             LOAD WCS
003645  0F73  E3C0 FC97                        LNJ    $B6,CSB20
003646                                  *
003647  0F75  0F83                             B      >IFT30
003648  0F76  E3C0 02C0                        LNJ    $B6,RDBK10            JUMP TO READ BACK SUBROUTINE
003649                                  *
003650                                  *
003651                                  *  STARTS TEST HERE
003652                                  *
003653  0F78  9B40 0523                 IFT30  LDR    $R1,SPHCOD            SPLASH MICROCODE
003654  0F7A  9F40 0006                        STR    $R1,IFT40
003655                                  *
003656                                  *
003657  0F7C  E3C0 FB1A                        LNJ    $B6,SRT210            ENABLE WCS AND STATUS CHECK
003658                                  *
003659  0F7E  8751                             CL     =$R1
003660  0F7F  FC80 1236                        LDB    $B7,<NULL             CLEAR B7
003661                                  *
003662  0F81  0000                      IFT40  RESV   1,0                    SPLASH.  ERROR IF O.K. NOT PRINTED
003663                                  *                                    IN 5 SEC
003664  0F82  1D00                             CMV    $R1,=0                R1 = 0?
003665  0F83  0981 001D                        BNE    IFT80                 NO.  REPORT ERROR
003666  0F85  FFC0 0505                        STB    $B7,TMP4-$AF+2        YES
003667  0F87  89C0 0503                        CMZ    TMP4+1                ERROR?
003668  0F89  0901 0017                        BE     IFT80                 YES.  REPORT ERROR
003669                                          CALL   ZV$TC,MSGOK           NO ERROR.  PRINT "O.K."
      0F8B  FBC0 0003
      0F8D  D380 0000       X
      0F8F  0F80
      0F90  1389
003670  0F91  0F81 017B                 EOP    B      EOP                   DONE
003671                                  *
003672                                  *
003673  0F93  ABC0 FC66                 IFT60  LAB    $B2,MC26              FETCH MICROCODE BRANCH TO 0FFB IF TRUE
003674  0F95  E3C0 FC6C                        LNJ    $B6,CSB10             FOR SEL REG SET FROM MAIN MEMORY AND
003675  0F97  E3C0 FC73                        LNJ    $B6,CSB20             PUT IT INTO WCS
003676                                  *
003677  0F99  ABC0 FC18                        LAB    $B2,MC8               FETCH MICROCODE INC R1, GO TO 0FFC
003678  0F9B  E3C0 FC66                        LNJ    $B6,CSB10             (RETURN ROUTINE) AND PUT IT INTO WCS
003679  0F9D  E3C0 FC6D                        LNJ    $B6,CSB20
003680  0F9F  0F81 FFC9                        B      IFT20
003681                                  *
003682                                  *
003683                                  *
003684                                  IFT80  CALL   ZV$ER,IF1,MSGIF1       ERROR "IF1"
      0FA1  FBC0 0003
      0FA3  D380 0000       X
      0FA5  0F80
      0FA6  0FAE
      0FA7  147F
003685                                          CALL   ZV$T,MSG4             PRINT "  * REPLACE MB *"
      0FA8  FBC0 0003
      0FAA  D380 0000       X
      0FAC  0F80
      0FAD  1319
003686  0FAE  0F7F                      IF1    NOP    >$-1                  **  ERROR  **
003687  0FAF  0F81 015D                        B      EOP                   DONE
003688                                  *
003689                                  *
003690                                  *
003691                                  *
003692                                  *
003693                                  *  TRANSPARENT MODE INTERFACE TEST
003694                                  **********************************
003695                                  *
003696                                  *
003697                                  *  MICROCODES FOR TRANSPARENT MODE TEST
003698                                  *
003699                                  *
003700  0FB1  13A0                      TMC1   DC     Z'13A0'               R7 = R3 + R7
003701  0FB2  9300                             DC     Z'9300'               GO TO RETURN ROUTINE (RAM2 PRESENT)
003702  0FB3  2000                             DC     Z'2000'
003703  0FB4  07FC                             DC     Z'07FC'
003704                                  *
003705  0FB5  13A8                      TMC2   DC     Z'13A8'               R7 =  R3 + R7 + 1
003706  0FB6  9300                             DC     Z'9300'               GO TO RETURN ROUTINE (RAM2 PRESENT)
003707  0FB7  2000                             DC     Z'2000'
003708  0FB8  07FC                             DC     Z'07FC'
003709                                  *
```

```
003710  0FB9  13A7              TMC3   DC    Z'13A7'          R7 = CPL R3 + R7
003711  0FBA  9F00                     DC    Z'9F00'          GO TO RETURN ROUTINE (RAM2 PRESENT)
003712  0FBB  2000                     DC    Z'2000'
003713  0FBC  07FC                     DC    Z'07FC'
003714                           *
003715  0FBD  03F3              TMC4   DC    Z'03F3'          R7 = R7 LEFT SHIFT 1
003716  0FBE  BF00                     DC    Z'BF00'          GO TO RETURN ROUTINE (RAM2 PRESENT)
003717  0FBF  2000                     DC    Z'2000'
003718  0FC0  07FC                     DC    Z'07FC'
003719                           *
003720  0FC1  0180              TMC5   DC    Z'0180'          Q <- D3
003721  0FC2  B300                     DC    Z'B300'
003722  0FC3  2000                     DC    Z'2000'
003723  0FC4  0001                     DC    Z'0001'
003724                           *
003725  0FC5  10B0              TMC6   DC    Z'10B0'          D0 <- D3
003726  0FC6  C300                     DC    Z'C300'
003727  0FC7  2000                     DC    Z'2000'
003728  0FC8  0002                     DC    Z'0002'
003729                           *
003730  0FC9  0000              TMC7   DC    Z'0000'          Q <-D0 + Q
003731  0FCA  8260                     DC    Z'8260'
003732  0FCB  0000                     DC    Z'0000'
003733  0FCC  0003                     DC    Z'0003'
003734                           *
003735  0FCD  01B0              TMC8   DC    Z'01B0'          D3 = D0 + Q
003736  0FCE  8300                     DC    Z'8300'
003737  0FCF  2000                     DC    Z'2000'
003738  0FD0  07FC                     DC    Z'07FC'
003739                           *
003740  0FD1  4F13              TMC9   DC    Z'4F13'          RAM8 <- ALU <- B7
003741  0FD2  BF00                     DC    Z'BF00'
003742  0FD3  2000                     DC    Z'2000'
003743  0FD4  0001                     DC    Z'0001'
003744                           *
003745  0FD5  CFFF              TMC10  DC    Z'CFFF'          ALU <- B1 <- RAM8
003746  0FD6  FBF0                     DC    Z'FBF0'          SHIFT LEFT <- XOR <- CPL B1
003747  0FD7  2000                     DC    Z'2000'
003748  0FD8  07FC                     DC    Z'07FC'
003749                           *
003750                           *
003751                           *
003752                           *
003753                           ************************************************
003754                           *
003755                           *
003756                           *
003757  0FD9  89C0 04B5          TIT10  CMZ   PASCNT+$AF-1     1ST PASS?
003758  0FDB  0981 000F                 BNE   TIT15            NO
003759  0FDD  E380 1225                 LNJ   $B6,<PRCRLF      YES.  PRINT C.R./L.F.
003760                                  CALL  ZV$TC,MSGSR      PRINT MESSAGE
        0FDF  FBC0 0003
        0FE1  D380 0000       X
        0FE3  0F80
        0FE4  13D2
003761                                  CALL  ZV$TC,MSGSQ
        0FE5  FBC0 0003
        0FE7  D380 0000       X
        0FE9  0F80
        0FEA  13AC
003762  0FEB  9870 0800          TIT15  LDR   $R1,=Z'0800'     SPLASH ADDR FOR RAM1
003763  0FED  9F40 04AD                 STR   $R1,SPHADR
003764  0FEF  9870 0080                 LDR   $R1,=Z'0080'     SPLASH MICROCODE FOR RAM1
003765  0FF1  9F40 04AA                 STR   $R1,SPHCOD
003766                           *
003767                           *
003768  0FF3  89C0 04A2                 CMZ   RAM2FL           RAM2 PRESENT?
003769  0FF5  0981 00F3                 BNE   TIT400           YES
003770  0FF7  ABC0 04D0                 LAB   $B2,BUFF3        RETURN ROUTINE (RAM1 ONLY)
003771  0FF9  9870 0BFF                 LDR   $R1,=Z'0BFF'
003772  0FFB  9F40 04A2                 STR   $R1,WCSIZE       WCS LAST LOCATION
003773  0FFD  9870 0BFC                 LDR   $R1,=Z'0BFC'     RETURN ROUTINE 1ST LOCATION
003774  0FFF  8051              TIT20  IO    =$R1,<OUTADR      WCS ADDR FOR RETURN ROUTINE
        1000  0000 14A2
003775  1002  2CF0                     LDV   $R2,=-16         TO COUNT 16 LOCATIONS
003776  1003  8072              TIT30  IO    +$B2,<OUTDAT      LOAD RETURN ROUTINE INTO WCS
        1004  0000 14A3
003777  1006  27FD                     BINC  $R2,>TIT30       BRANCH IF ALL LOCATIONS NOT LOADED
003778                           *
003779                           *
003780  1007  8740 0498                 CL    TSTCNT           CLEAR TEST COUNTER
003781                           *                            IF O.K. NOT PRINTED IN 5 SEC. THE COUNTER
003782                           *                            NO. SHOWS WHICH TEST THAT FAILS
003783                           *
003784                           *
003785                           *  TEST 1
003786                           *
003787                           *
003788  1009  0F7F                     NOP   >$-1
003789                           *
003790                           *
003791  100A  8000 149B                 IO    <SPHADR,<OUTADR  WCS 1ST ENTRY  LOCATION
        100C  0000 14A2
003792  100E  8AC0 0491                 INC   TSTCNT           BUMP TEST COUNTER
003793  1010  ABC0 FFA0                 LAB   $B2,IMC1         FETCH MICROCODE R7 = R3 + R7
003794  1012  E3C0 00F2                 LNJ   $B6,TIT500       CHECK FOR RAM2 PRESENT
003795  1014  E3C0 FBED                 LNJ   $B6,CSB10        LOAD WCS
003796  1016  E3C0 FBF4                 LNJ   $B6,CSB20
003797                           *
003798                           *  STARTS TEST HERE
003799                           *
003800  1018  9840 0483                 LDR   $R1,SPHCOD       SPLASH MICROCODE
003801  101A  9F40 0005                 STR   $R1,TIT40
003802  101C  3C05                     LDV   $R3,=5           INITIALIZE R3
003803  101D  7C05                     LDV   $R7,=5           INITIALIZE R7
003804                           *
003805                           *
003806  101E  E3C0 FA78                 LNJ   $B6,SRT210       ENABLE WCS AND STATUS CHECK
003807                           *
003808                           *
003809  1020  0000              TIT40  RESV  1,0              SPLASH.  ERROR IF O.K. NOT PRINTED
003810                           *                            IN 5 SEC
003811  1021  7D0A                     CMV   $R7,=10          ERROR?
```

```
003812  1022  0981 00D0          BNE   TIT460            YES.  REPORT ERROR
003813                      *
003814                      *
003815                      *
003816                      *
003817                      *
003818                      *  TEST 2
003819                      *
003820                      *
003821  1024  0F7F                NOP   >$-1
003822                      *
003823                      *
003824  1025  8000 149B           IO    <SPHADR,<OUTADR     WCS 1ST ENTRY LOCATION
        1027  0000 14A2
003825  1029  8AC0 0476           INC   TSTCNT              BUMP TEST COUNTER
003826  102B  ABC0 FF89           LAB   $B2,IMC2            FETCH MICROCODE R7 = R7 + R3 + 1
003827  102D  E3C0 00D7           LNJ   $B6,TIT500          CHECK FOR RAM2 PRESENT
003828  102F  E3C0 FBD2           LNJ   $B6,CSB10           LOAD WCS
003829  1031  E3C0 FBD9           LNJ   $B6,CSB20
003830                      *
003831                      *  STARTS TEST HERE
003832                      *
003833  1033  9840 0468           LDR   $R1,SPHCOD          SPLASH MICROCODE
003834  1035  9F40 0005           STR   $R1,TIT50
003835  1037  3C05                LDV   $R3,=5              INITIALIZE R3
003836  1038  7C05                LDV   $R7,=5              INITIALIZE R7
003837                      *
003838                      *
003839  1039  E3C0 FA5D           LNJ   $B6,SRT210          ENABLE WCS AND STATUS CHECK
003840                      *
003841                      *
003842  103B  0000          TIT50 RESV  1,0                 SPLASH.  ERROR IF O.K. NOT PRINTED
003843                      *                               IN 5 SEC
003844  103C  7D0B                CMV   $R7,=11             ERROR?
003845  103D  0981 00B5           BNE   TIT460              YES.  REPORT ERROR
003846                      *
003847                      *
003848                      *
003849                      *  TEST 3
003850                      *
003851                      *
003852  103F  0F7F                NOP   >$-1
003853                      *
003854                      *
003855  1040  8000 149B           IO    <SPHADR,<OUTADR     WCS 1ST ENTRY LOCATION
        1042  0000 14A2
003856  1044  8AC0 045B           INC   TSTCNT              BUMP TEST COUNTER
003857  1046  ABC0 FF72           LAB   $B2,IMC3            FETCH MICROCODE R7 = CPL R3 + R7
003858  1048  E3C0 00BC           LNJ   $B6,TIT500          CHECK FOR RAM2 PRESENT
003859  104A  E3C0 FBB7           LNJ   $B6,CSB10           LOAD WCS
003860  104C  E3C0 FBBE           LNJ   $B6,CSB20
003861                      *
003862                      *  STARTS TEST HERE
003863                      *
003864  104E  9840 044D           LDR   $R1,SPHCOD          SPLASH MICROCODE
003865  1050  9F40 0005           STR   $R1,TIT60
003866  1052  3C05                LDV   $R3,=5              INITIALIZE R3
003867  1053  7C05                LDV   $R7,=5              INITIALIZE R7
003868                      *
003869                      *
003870  1054  E3C0 FA42           LNJ   $B6,SRT210          ENABLE WCS AND STATUS CHECK
003871                      *
003872                      *
003873  1056  0000          TIT60 RESV  1,0                 SPLASH.  ERROR IF O.K. NOT  PRINTED
003874                      *                               IN 5 SEC
003875  1057  7DFF                CMV   $R7,=-1             ERROR?
003876  1058  0981 009A           BNE   TIT460              YES.  REPORT ERROR
003877                      *
003878                      *
003879                      *  TEST 4
003880                      *
003881                      *
003882  105A  0F7F                NOP   >$-1
003883                      *
003884                      *
003885  105B  8000 149B           IO    <SPHADR,<OUTADR     WCS 1ST ENTRY LOCATION
        105D  0000 14A2
003886  105F  8AC0 0440           INC   TSTCNT              BUMP TEST COUNTER
003887  1061  ABC0 FF5B           LAB   $B2,IMC4            FETCH MICROCODE R7 = R7 LEFT SHIFT 1
003888  1063  E3C0 00A1           LNJ   $B6,TIT500          CHECK FOR RAM2 PRESENT
003889  1065  E3C0 FB9C           LNJ   $B6,CSB10           LOAD WCS
003890  1067  E3C0 FBA3           LNJ   $B6,CSB20
003891                      *
003892                      *  STARTS TEST HERE
003893                      *
003894  1069  9840 0432           LDR   $R1,SPHCOD          SPLASH MICROCODE
003895  106B  9F40 0004           STR   $R1,TIT70
003896  106D  7C05                LDV   $R7,=5              INITIALIZE R7
003897                      *
003898                      *
003899  106E  E3C0 FA28           LNJ   $B6,SRT210          ENABLE WCS AND STATUS CHECK
003900                      *
003901                      *
003902  1070  0000          TIT70 RESV  1,0                 SPLASH.  ERROR IF O.K. NOT PRINTED
003903                      *                               IN 5 SEC
003904  1071  7D0A                CMV   $R7,=10             ERROR?
003905  1072  0981 0080           BNE   TIT460              YES.  REPORT ERROR
003906                      *
003907                      *
003908                      *
003909                      *  TEST 5
003910                      *
003911                      *
003912  1074  0F7F                NOP   >$-1
003913                      *
003914                      *
003915  1075  8000 149B           IO    <SPHADR,<OUTADR     WCS 1ST LOCATION
        1077  0000 14A2
003916  1079  8AC0 0426           INC   TSTCNT              BUMP TEST COUNTER
003917  107B  ABC0 FF45           LAB   $B2,IMC5            FETCH MICROCODE Q <- R3
003918  107D  E3C0 FB84           LNJ   $B6,CSB10           LOAD WCS
003919  107F  E3C0 FB8B           LNJ   $B6,CSB20
003920                      *
```

```
003921                                  *
003922  1081  ABC0 FF43                        LAB   $B2,IMC6          FETCH MICROCODE R0 <- R3
003923  1083  E3C0 FB7E                        LNJ   $B6,CSB10         LOAD WCS
003924  1085  E3C0 FB85                        LNJ   $B6,CSB20
003925                                  *
003926                                  *
003927  1087  ABC0 FF41                        LAB   $B2,IMC7          FETCH MICROCODE Q <- R0 + Q
003928  1089  E3C0 FB78-                       LNJ   $B6,CSB10         LOAD WCS
003929  108B  E3C0 FB7F                        LNJ   $B6,CSB20
003930                                  *
003931                                  *
003932  108D  ABC0 FF3F                        LAB   $B2,IMC8          FETCH MICROCODE R3 = R0 + Q
003933  108F  E3C0 0075                        LNJ   $B6,IIT500        CHECK FOR RAM2 PRESENT
003934  1091  E3C0 FB70                        LNJ   $B6,CSB10         LOAD WCS
003935  1093  E3C0 FB77                        LNJ   $B6,CSB20
003936                                  *
003937                                  *  STARTS TEST HERE
003938                                  *
003939  1095  9840 0406                        LDR   $R1,SPHCOD        SPLASH MICROCODE
003940  1097  9F40 0004                        STR   $R1,IIT80
003941  1099  3C05                            LDV   $R3,=5            INITIALIZE R3
003942                                  *
003943                                  *
003944  109A  E3C0 F9FC                        LNJ   $B6,SRT210        ENABLE WCS AND STATUS CHECK
003945                                  *
003946  109C  0000                      IIT80 RESV  1,0               SPLASH.  ERROR IF 0. K. NOT PRINTED
003947                                  *                             IN 5 SEC
003948                                  *
003949  109D  3D0F                            CMV   $R3,=15           ERROR?
003950  109E  0981 0054                        BNE   IIT460            YES.  REPORT ERROR
003951                                  *
003952                                  *
003953                                  *  TEST 6
003954                                  *
003955                                  *
003956  10A0  0F7F                            NOP   >$-1
003957                                  *
003958                                  *
003959  10A1  8000 149B                        IO    <SPHADR,<OUTADR   WCS 1ST ENTRY LOCATION
        10A3  0000 14A2
003960  10A5  8AC0 03FA                        INC   TSTCNT            BUMP TEST COUNTER
003961  10A7  ABC0 FF29                        LAB   $B2,IMC9          FETCH MICROCODE RAM8 <- ALU <- B7
003962  10A9  E3C0 FB58                        LNJ   $B6,CSB10         LOAD WCS
003963  10AB  E3C0 FB5F                        LNJ   $B6,CSB20
003964                                  *
003965  10AD  ABC0 FF27                        LAB   $B2,IMC10         FETCH MICROCODE
003966  10AF  E3C0 0055                        LNJ   $B6,IIT500        CHECK FOR RAM2 PRESENT
003967  10B1  E3C0 FB50                        LNJ   $B6,CSB10         LOAD WCS
003968  10B3  E3C0 FB57                        LNJ   $B6,CSB20
003969                                  *
003970                                  *  STARTS TEST HERE
003971                                  *
003972  10B5  9840 03E6                        LDR   $R1,SPHCOD        SPLASH MICROCODE
003973  10B7  9F40 0013                        STR   $R1,IIT90
003974  10B9  1C05                            LDV   $R1,=5
003975  10BA  8740 03CF                        CL    TMP4
003976  10BC  9F40 03CE                        STR   $R1,TMP4+1
003977  10BE  FCC0 03CC                        LDB   $B7,TMP4-$AF+2    INITIALIZE B7
003978  10C0  1CFF                            LDV   $R1,=-1
003979  10C1  9F40 03C8                        STR   $R1,TMP4
003980  10C3  9870 FFF4                        LDR   $R1,=Z'FFF4'
003981  10C5  9F40 03C5                        STR   $R1,TMP4+1
003982  10C7  ACC0 03C3                        LDB   $B2,TMP4-$AF+2    DATA TO COMPARE
003983                                  *
003984                                  *
003985  10C9  E3C0 F9CD                        LNJ   $B6,SRT210        ENABLE WCS AND STATUS CHECK
003986                                  *
003987  10CB  0000                      IIT90 RESV  1,0               SPLASH.  ERROR IF 0. K. NOT PRINTED
003988                                  *                             IN 5 SEC
003989                                  *
003990  10CC  9840 0000       P                 LDR   $R1,ZV$AF
003991  10CE  1D02                              CMV   $R1,=2            LAF?
003992  10CF  090A                              BE    >IIT100           YES
003993  10D0  FFC0 03B9                         STB   $B7,TMP4          NO
003994  10D2  9840 03B7                         LDR   $R1,TMP4
003995  10D4  9970 FFF4                         CMR   $R1,=Z'FFF4'      MATCH?
003996  10D6  0901 0005                         BE    IIT110            YES
003997  10D8  0F9B                             B     >IIT460           NO.  REPORT ERROR
003998  10D9  FDD2                      IIT100 CMB   $B7,=$B2          MATCH?
003999  10DA  0981 0018                         BNE   IIT460            NO.  REPORT ERROR
004000  10DC  0681 0016                 IIT110 BCF   IIT460            REPORT ERROR IF CARRY BIT NOT  SET
004001                                  *
004002                                  *
004003                                  *
004004                                  *  TEST 7
004005                                  *
004006                                  *
004007  10DE  0F7F                              NOP   >$-1
004008                                  *
004009                                  *
004010                                  *  ZERO NA FIELD TEST
004011                                  *
004012                                  *
004013  10DF  0F81 FDA4                         B     ZNT10             BRANCH TO ZERO NA FIELD TEST
004014                                  *
004015                                  *
004016                                  *
004017                                  IIT390 CALL  ZV$TC,MSGOK        NO ERROR.  PRINT "O.K."
        10E1  FBC0 0003
        10E3  D380 0000       X
        10E5  0F80
        10E6  1389
004018                                  *
004019                                  *
004020  10E7  0F81 0025                         B     EOP               DONE
004021                                  *
004022                                  *
004023                                  *
004024                                  *
004025                                  *
004026                                  *
004027                                  *
004028  10E9  ABC0 03EE                 IIT400 LAB   $B2,BUFF4         RETURN ROUTINE (RAM2 PRESENT)
```

```
004029  10EB  9870 0FFF              LDR   $R1,=Z'0FFF'
004030  10ED  9F40 03B0              STR   $R1,WCSIZE          WCS LAST LOCATION
004031  10EF  9870 0FFC              LDR   $R1,=Z'0FFC'
004032  10F1  0F81 FF0D              B     TIT20
004033                          *
004034                          *
004035                          *
004036                          *
004037  10F3  D840 03AC          TIT460 LDR   $R5,TSTCNT          R5 SHOWS WHICH TEST THAT FAILS
004038                                  CALL  ZV$ER,TI1,MSGTI1    ERROR "TI1"
        10F5  FBC0 0003
        10F7  D380 0000       X
        10F9  0F80
        10FA  1102
        10FB  1481
004039                                  CALL  ZV$T,MSG4           PRINT "  * REPLACE MB *"
        10FC  FBC0 0003
        10FE  D380 0000       X
        1100  0F80
        1101  1319
004040  1102  0F7F              TI1    NOP   >$-1                ** ERROR **
004041                          *                                TRANSPARENT MODE INTERFACE ERROR
004042  1103  0F81 0009                 B     EOP
004043                          *
004044                          *
004045                          *
004046                          *
004047                          *  SUBROUTINE TO MODIFY MICROCODE'S NEXT ADDR FIELD GOING TO RETURN ROUTINE
004048                          *  WHEN RAM2 IS PRESENT
004049                          *
004050  1105  89C0 0390          TIT500 CMZ   RAM2FL              RAM2 PRESENT?
004051  1107  0985                      BNE   >TIT510             YES
004052  1108  9870 0BFC                 LDR   $R1,=Z'0BFC'        NO.  MODIFY ADDR TO GO TO RETURN ROUTINE
004053  110A  9F42 0003                 STR   $R1,$B2.3
004054  110C  8366              TIT510 JMP   $B6                 RETURN
004055                          *
004056                          *
004057                          *
004058                          *
004059                          *
004060                          *
004061                          *
004062                          * END-OF-PASS
004063                          ************
004064                          *
004065                          *
004066                          *
004067  110D  8A80 148F          EOP    INC   <PASCNT+$AF-1       BUMP PASS COUNT
004068                          *
004069                          * DISPLAY PASS COUNT IN B4 IN DECIMAL
004070                          *
004071  110F  F800 148F                 LDR   $R7,<PASCNT+$AF-1
004072  1111  A870 0003                 LDR   $R2,=3
004073  1113  8756              DIGIT  CL    =$R6
004074  1114  F370 000A                 DIV   $R7,=10
004075  1116  EAD5                      SRM   $R6,=$R5,=Z'000F'
        1117  000F
004076  1118  5054                      SCR   $R5,4
004077  1119  277A                      BDEC  $R2,>DIGIT
004078  111A  DF00 1490                 STR   $R5,<B4DCML+$AF-1
004079  111C  CC80 1490                 LDB   $B4,<B4DCML         **  B4 RESERVED FOR PASS COUNT DISPLAY  **
004080                          *
004081                          * TYPE PASS COUNT AND "3 MIN DONE" IF 3 MINUTES ARE UP
004082                          *
004083                          *
004084  111E  9800 0000       X        LDR   $R1,<ZHRTCC         ARE 3 MINUTES UP?
004085  1120  9970 AB9F                 CMR   $R1,=Z'AB9F'        AB9F = 43935(DEC) = 65535-21600(DEC).
004086                          * CLOCK SET INITIALLY TO FFFF=65535(DEC).  21600 TICKS = 3 MINUTES.
004087  1122  031B                      BG    >GLEN               3 MINUTES NOT UP YET
004088                          *
004089  1123  1CFF                      LDV   $R1,=-1             SET UP FOR NEXT 3 MIN INTERVAL
004090  1124  9F00 0000       X        STR   $R1,<ZHRTCC
004091                                  CALL  ZV$TC,MSG9          TYPE PASS MESSAGE
        1126  FBC0 0003
        1128  D380 0000       X
        112A  0F80
        112B  141A
004092                                  CALL  ZV$TD,PASCNT+$AF-1  TYPE THE COUNT
        112C  FBC0 0003
        112E  D380 0000       X
        1130  0F80
        1131  148F
004093                                  CALL  ZV$TC,MSG10         TYPE "3 MIN DONE"
        1132  FBC0 0003
        1134  D380 0000       X
        1136  0F80
        1137  141F
004094  1138  0004                      RTCN
004095  1139  8700 1493                 CL    <RED                CLEAR RED ERROR BEFORE QUIT
004096                          *
004097  113B  0F80 0107                 B     <DONE               **  TESTING COMPLETE  **
004098                          *
004099                          * TYPE PASS COUNT
004100                          *
004101                          GLEN   CALL  ZV$TC,MSG9           TYPE PASS MESSAGE
        113D  FBC0 0003
        113F  D380 0000       X
        1141  0F80
        1142  141A
004102                                  CALL  ZV$TD,PASCNT+$AF-1  TYPE THE COUNT
        1143  FBC0 0003
        1145  D380 0000       X
        1147  0F80
        1148  148F
004103  1149  0F80 0135                 B     <NXTPAS             REPEAT TEST, DO NEXT PASS
004104                          *
004105                          *
004106                          *
004107                          *
004108                          *  S U B R O U T I N E S
004109                          *
004110                          *
004111                          * NAK ERROR   (WCS MOTHER BOARD FAULT)
```

```
004112                                  *
004113                                  *   CALLING SEQUENCE:
004114                                  *      LNJ   $B6,<NAKERR
004115                                  *
004116  114B  9F40 0033                NAKERR STR   $R1,NAKER4          SAVE $R1
004117  114D  EFC0 000D                       STB   $B6,NAKER2
004118  114F  88C0 000B                       DEC   NAKER2+$AF-1        ADJUST REPORTING ADRS
004119  1151  88C0 0009                       DEC   NAKER2+$AF-1
004120  1153  9800 0000       X               LDR   $R1,<ZV$AF
004121  1155  1B83                           BODD  $R1,>NAKER1         (ZV$AF=1 IF SAF)
004122  1156  88C0 0004                       DEC   NAKER2+$AF-1        DECREMENT AGAIN IF LAF (ZV$AF=2)
004123                                  *
004124  1158  D380 0000       X        NAKER1 LNJ   $B5,<ZV$ER          ERROR "NAK"
004125  115A  0F83                           B     >NAKER3             RETURN FROM ZV$ER
004126  115B  0000                     NAKER2 RESV  $AF,0               REPORTING ADRS - I/O INSTRUCTION
004127                                  *                               PRECEDING THIS ADRS CAUSED ERROR
004128  115C  1315                           DC    <MSG2               "NAK"
004129  115D  9840 0021                NAKER3 LDR   $R1,NAKER4          RESTORE $R1
004130                                  *
004131                                         CALL  ZV$T,MSG4          PRINT "   * REPLACE MB *"
        115F  FBC0 0003
        1161  D380 0000       X
        1163  0F80
        1164  1319
004132  1165  8040 0019                       IO    NAKER4,<INSTAT     INPUT STATUS
        1167  0000 14A6
004133  1169  0709                           BIOT  >NKER3B
004134                                  NKER3A CALL  ZV$ER,NKER3A,MSG2   ERROR 'NAK'
        116A  FBC0 0003
        116C  D380 0000       X
        116E  0F80
        116F  116A
        1170  1315
004135  1171  0F8D                           B     >NKER3C             INPUT STATUS SHOULD NEVER BE NAK'D.
004136                                  *                               REPLACE WCS MOTHER BOARD.
004137                                  *
004138                                  NKER3B CALL  ZV$T,MSG11          PRINT "   STATUS ="
        1172  FBC0 0003
        1174  D380 0000       X
        1176  0F80
        1177  1425
004139                                         CALL  ZV$THZ,NAKER4       PRINT STATUS READ
        1178  FBC0 0003
        117A  D380 0000       X
        117C  0F80
        117D  117F
004140                                  *
004141  117E  8386                     NKER3C JMP   $B6                 RETURN
004142                                  *
004143  117F  0000                     NAKER4 RESV  1,0                 SAVE AREA FOR R1 AND STATUS
004144                                  *
004145                                  *
004146                                  * ACK ERROR   (WCS MOTHER BOARD FAULT)
004147                                  *
004148                                  *   CALLING SEQUENCE:
004149                                  *      LNJ   $B6,<ACKERR
004150                                  *
004151  1180  9F40 0033                ACKERR STR   $R1,ACKER4          SAVE $R1
004152  1182  EFC0 000D                       STB   $B6,ACKER2
004153  1184  88C0 000B                       DEC   ACKER2+$AF-1        ADJUST REPORTING ADRS
004154  1186  88C0 0009                       DEC   ACKER2+$AF-1
004155  1188  9800 0000       X               LDR   $R1,<ZV$AF
004156  118A  1B83                           BODD  $R1,>ACKER1         (ZV$AF=1 IF SAF)
004157  118B  88C0 0004                       DEC   ACKER2+$AF-1        DECREMENT AGAIN IF LAF (ZV$AF=2)
004158                                  *
004159  118D  D380 0000       X        ACKER1 LNJ   $B5,<ZV$ER          ERROR "ACK"
004160  118F  0F83                           B     >ACKER3             RETURN FROM ZV$ER
004161  1190  0000                     ACKER2 RESV  $AF,0               REPORTING ADRS - I/O INSTRUCTION
004162                                  *                               PRECEDING THIS ADRS CAUSED ERROR.
004163  1191  1317                           DC    <MSG3               "ACK"
004164  1192  9840 0021                ACKER3 LDR   $R1,ACKER4          RESTORE $R1
004165                                  *
004166                                         CALL  ZV$T,MSG4          PRINT "   * REPLACE MB *"
        1194  FBC0 0003
        1196  D380 0000       X
        1198  0F80
        1199  1319
004167  119A  8040 0019                       IO    ACKER4,<INSTAT     INPUT STATUS
        119C  0000 14A6
004168  119E  0709                           BIOT  >AKER3B
004169                                  AKER3A CALL  ZV$ER,AKER3A,MSG2   ERROR "ACK"
        119F  FBC0 0003
        11A1  D380 0000       X
        11A3  0F80
        11A4  119F
        11A5  1315
004170  11A6  0F8D                           B     >AKER3C             INPUT STATUS SHOULD NEVER BE NAK'D.
004171                                  *                               REPLACE WCS MOTHER BOARD.
004172                                  *
004173                                  AKER3B CALL  ZV$T,MSG11          PRINT "   STATUS ="
        11A7  FBC0 0003
        11A9  D380 0000       X
        11AB  0F80
        11AC  1425
004174                                         CALL  ZV$THZ,ACKER4       PRINT STATUS READ
        11AD  FBC0 0003
        11AF  D380 0000       X
        11B1  0F80
        11B2  11B4
004175                                  *
004176  11B3  8386                     AKER3C JMP   $B6                 RETURN
004177                                  *
004178  11B4  0000                     ACKER4 RESV  1,0                 SAVE AREA FOR R1 AND STATUS
004179                                  *
004180                                  *
004181                                  * STATUS ERROR   (WCS MOTHER BOARD FAULT)
004182                                  *
004183                                  *   CALLING SEQUENCE:
004184                                  *      LNJ   $B6,<STERR
004185                                  *   STATUS READ IS IN R4 (IS)
004186                                  *   STATUS "SHOULD BE" IS IN R5 (SB)
004187                                  *      (IF R5=FFFF, DON'T PRINT STATUS SB)
004188                                  *
```

```
004189  11B5  9F40 0039          STERR  STR   $R1,STERR4        SAVE $R1
004190  11B7  EFC0 000F                 STB   $B6,STERR2
004191  11B9  88C0 000D                 DEC   STERR2+$AF-1      ADJUST REPORTING ADRS
004192  11BB  88C0 000B                 DEC   STERR2+$AF-1
004193  11BD  9800 0000       X         LDR   $R1,<ZV$AF
004194  11BF  1B83                      BODD  $R1,>STERR1       (ZV$AF=1 IF SAF)
004195  11C0  88C0 0006                 DEC   STERR2+$AF-1      DECREMENT AGAIN IF LAF (ZV$AF=2)
004196  11C2  9840 002C          STERR1 LDR   $R1,STERR4        RESTORE $R1
004197                           *
004198  11C4  D380 0000       X         LNJ   $B5,<ZV$ER        ERROR "STATUS"
004199  11C6  0F83                      B     >STERR3           RETURN FROM ZV$ER
004200  11C7  0000               STERR2 RESV  $AF,0             REPORTING ADRS - REFER TO THIS ADRS
004201                                                          IN LISTING FOR COMMENT.
004202  11C8  1385                      DC    <MSG5             "STATUS"
004203                           *
004204                           STERR3 CALL  ZV$T,MSG6         PRINT "   IS"
        11C9  FBC0 0003
        11CB  D380 0000       X
        11CD  0F80
        11CE  1412
004205  11CF  CF40 001F                 STR   $R4,STERR4
004206                                  CALL  ZV$THZ,STERR4     PRINT STATUS READ
        11D1  FBC0 0003
        11D3  D380 0000       X
        11D5  0F80
        11D6  11EF
004207  11D7  D970 FFFF                 CMR   $R5,=Z'FFFF'
004208  11D9  090F                      BE    >STER3A           DON'T PRINT STATUS SB
004209                           *
004210                                  CALL  ZV$T,MSG7         PRINT ",  SB"
        11DA  FBC0 0003
        11DC  D380 0000       X
        11DE  0F80
        11DF  1415
004211  11E0  DF40 000E                 STR   $R5,STERR4
004212                                  CALL  ZV$THZ,STERR4     PRINT STATUS SHOULD BE
        11E2  FBC0 0003
        11E4  D380 0000       X
        11E6  0F80
        11E7  11EF
004213                           STER3A CALL  ZV$T,MSG4         PRINT "   * REPLACE MB *"
        11E8  FBC0 0003
        11EA  D380 0000       X
        11EC  0F80
        11ED  1319
004214                           *
004215  11EE  8386                      JMP   $B6               RETURN
004216  11EF  0000               STERR4 RESV  1,0               TEMP STORE FOR R1,R4,R5
004217                           *
004218                           *
004219                           * SUBROUTINE TO PRINT MSG4
004220                           *
004221                           *   CALLING SEQUENCE:
004222                           *      LNJ   $B6,<PRRMB
004223                           *
004224                           PRRMB  CALL  ZV$T,MSG4         PRINT "   * REPLACE MB *"
        11F0  FBC0 0003
        11F2  D380 0000       X
        11F4  0F80
        11F5  1319
004225  11F6  8386                      JMP   $B6               RETURN
004226                           *
004227                           *
004228                           * SUBROUTINE TO PRINT MSG4A
004229                           *
004230                           *   CALLING SEQUENCE:
004231                           *      LNJ   $B6,<PRRDB1
004232                           *
004233                           PRRDB1 CALL  ZV$T,MSG4A        PRINT "   * REPLACE DB 1 *"
        11F7  FBC0 0003
        11F9  D380 0000       X
        11FB  0F80
        11FC  1322
004234  11FD  8386                      JMP   $B6               RETURN
004235                           *
004236                           *
004237                           * SUBROUTINE TO PRINT MSG4B
004238                           *
004239                           *   CALLING SEQUENCE:
004240                           *      LNJ   $B6,<PRRDB2
004241                           *
004242                           PRRDB2 CALL  ZV$T,MSG4B        PRINT "   * REPLACE DB 2 *"
        11FE  FBC0 0003
        1200  D380 0000       X
        1202  0F80
        1203  132C
004243  1204  8386                      JMP   $B6               RETURN
004244                           *
004245                           *
004246                           * SUBROUTINE TO PRINT MSG4C
004247                           *
004248                           *   CALLING SEQUENCE:
004249                           *      LNJ   $B6,<PRMAY
004250                           *
004251                           PRMAY  CALL  ZV$T,MSG4C        PRINT "   * MAY BE WCS MB *"
        1205  FBC0 0003
        1207  D380 0000       X
        1209  0F80
        120A  1336
004252  120B  8386                      JMP   $B6               RETURN
004253                           *
004254                           *
004255                           * SUBROUTINE TO PRINT STATUS
004256                           *
004257                           *   CALLING SEQUENCE:
004258                           *      LNJ   $B6,<PRSTS
004259                           *
004260                           PRSTS  CALL  ZV$T,MSG11        PRINT "   STATUS ="
        120C  FBC0 0003
        120E  D380 0000       X
        1210  0F80
        1211  1425
004261  1212  CF40 0010                 STR   $R4,PRSTS1        SAVE R4
```

```
004262  1214  8754                        CL    =$R4
004263  1215  8054                        IO    =$R4,<INSTAT
        1216  0000 14A6
004264  1218  CF40 000B                    STR   $R4,PRSTS2
004265                                     CALL  ZV$THZ,PRSTS2          PRINT STATUS
        121A  FBC0 0003
        121C  D380 0000       X
        121E  0F80
        121F  1224
004266  1220  C840 0002                    LDR   $R4,PRSTS1             RESTORE R4
004267  1222  8386                        JMP   $B6                    RETURN
004268                              *
004269  1223  0000                 PRSTS1 RESV  1,0                    TEMP SAVE
004270  1224  0000                 PRSTS2 RESV  1,0                    TEMP SAVE
004271                              *
004272                              *
004273                              *
004274                              *  SUBROUTINE TO PRINT CR/LF
004275                              *
004276                              *   CALLING SEQUENCE:
004277                              *       LNJ   $B6,<PRCRLF
004278                              *
004279  1225  8F40 000C             PRCRLF SAVE  PRCLSV,=Z'0005'        SAVE B5,B7
        1227  0005
004280                                     CALL  ZV$TC,NULL             PRINT CARRIAGE RETURN/LINE FEED
        1228  FBC0 0003
        122A  D380 0000       X
        122C  0F80
        122D  1236
004281  122E  8FC0 0003                    RSTR  PRCLSV,=Z'0005'        RESTORE B5,B7
        1230  0005
004282  1231  8386                        JMP   $B6                    RETURN
004283  1232  0000                 PRCLSV RESV  4,0                    SAVE AREA FOR B5,B7
004284                              *
004285                              *
004286  1236  0000                 NULL   RESV  $AF,0
004287                              *
004288                              *
004289                              *  OFF LINE READ BACK FROM WCS
004290                              *
004291  1237  8753                 RDBK10 CL    =$R3                    INDEX
004292  1238  89C0 025D                    CMZ   RAM2FL                 RAM2  PRESENT?
004293  123A  0992                        BNE   >RDBK40                YES
004294  123B  C870 0FFF                    LDR   $R4,=Z'0FFF'           NO.  4K WORDS
004295  123D  BC80 0000       X     RDBK15 LDB   $B3,<ZV$LR             MEMORY LOW
004296  123F  8000 149B             RDBK20 IO    <SPHADR,<OUTADR         SPLASH ADDR
        1241  0000 14A2
004297  1243  07FC                        BIOF  >RDBK20
004298  1244  8056                 RDBK30 IO    =$R6,<INDAT             READ WCS
        1245  0000 14A5
004299  1247  07FD                        BIOF  >RDBK30
004300  1248  EF7F                        STR   $R6,$B3,+$R3           STORE WCS READ OUT MICROCODES
004301  1249  B954                        CMR   $R3,=$R4               ALL MICROCODES STORED?
004302  124A  03FA                        BLE   >RDBK30                NO
004303  124B  8386                        JMP   $B6                    YES
004304  124C  C870 1FFF             RDBK40 LDR   $R4,=Z'1FFF'           RAM2 PRESENT.  8K WORDS
004305  124E  0FEF                        B     >RDBK15
004306                              *
004307                              *
004308                              *  OFF LINE PRINT MICROCODES IN WCS
004309                              *
004310  124F  8740 024D                    CL    RTFLG                  RETURN ROUTINE FLAG
004311  1251  8753                 PRNT10 CL    =$R3                   INDEX
004312  1252  BCC0 0000       P           LDB   $B3,ZV$LR              MEMORY LOW
004313  1254  0F7F                        NOP   >$-1                   B3 = ZV$LR IS THE 1ST LOCATION IN WCS
004314                              *                                  CAN CHANGE B3 TO START PRINTING FOR
004315                              *                                  LOCATIONS OTHER THAN 1ST WCS LOCATION
004316  1255  89C0 0240                    CMZ   RAM2FL                 RAM2 PRESENT?
004317  1257  099A                        BNE   >PRNT40                YES
004318  1258  E870 0FFF                    LDR   $R6,=Z'0FFF'           NO.  4K WORDS
004319  125A  89C0 0242                    CMZ   RTFLG                  RETURN ROUTINE FLAG SET?
004320  125C  0981 0020                    BNE   PRNT50                 YES.  PRINT RETURN ROUTINE
004321  125E  E380 1225             PRNT15 LNJ   $B6,<PRCRLF            PRINT C.R./L.F.
004322  1260  5CFC                        LDV   $R5,=-4
004323  1261  C873                 PRNT20 LDR   $R4,+$B3                READ
004324  1262  CF40 000D                    STR   $R4,PRNT30             MICROCODE
004325                                     CALL  ZV$THZ,PRNT30          PRINT MICROCODE
        1264  FBC0 0003
        1266  D380 0000       X
        1268  0F80
        1269  1270
004326  126A  8AD3                        INC   =$R3                   BUMP INDEX
004327  126B  B956                        CMR   $R3,=$R6               ALL MICROCODES PRINTED?
004328  126C  0903                        BE    >PRNT25                YES
004329  126D  57F4                        BINC  $R5,>PRNT20            NO.
004330  126E  0FF0                        B     >PRNT15
004331  126F  0000                 PRNT25 HLT
004332  1270  0000                 PRNT30 RESV  1,0
004333  1271  E870 1FFF             PRNT40 LDR   $R6,=Z'1FFF'           RAM2 PRESENT.  8K WORDS
004334                              *
004335  1273  89C0 0229                    CMZ   RTFLG                  RETURN ROUTINE FLAG SET?
004336  1275  0981 0015                    BNE   PRNT60                 YES.  PRINT RETURN ROUTINE
004337  1277  0F81 FFE6                    B     PRNT15                 NO
004338                              *
004339                              *
004340                              *  PRINT RETURN ROUTINE
004341                              *
004342  1279  8AC0 0223                    INC   RTFLG                  SET RETURN ROUTINE FLAG
004343  127B  0F81 FFD5                    B     PRNT10
004344                              *
004345  127D  BFC0 020D             PRNT50 STB   $B3,TMP4-$AF+2         CALCULATE RETURN ROUTINE ADDR
004346  127F  B840 020B                    LDR   $R3,TMP4+1
004347  1281  BA70 0FF0                    ADD   $R3,=Z'0FF0'
004348  1283  BF40 0207                    STR   $R3,TMP4+1
004349  1285  B870 0FF0                    LDR   $R3,=Z'0FF0'
004350  1287  BCC0 0203             PRNT55 LDB   $B3,TMP4-$AF+2
004351  1289  0F81 FFD4                    B     PRNT15
004352                              *
004353  128B  BFC0 01FF             PRNT60 STB   $B3,TMP4-$AF+2         CALCULATE RETURN ROUTINE ADDR
004354  128D  B840 01FD                    LDR   $R3,TMP4+1
004355  128F  BA70 1FF0                    ADD   $R3,=Z'1FF0'
004356  1291  BF40 01F9                    STR   $R3,TMP4+1
004357  1293  B870 1FEF                    LDR   $R3,=Z'1FEF'
```

```
004358  1295  0FF2                        B     >PRNT55
004359                                  *
004360                                  *
004361                                  *
004362                                  *
004363                                  *
004364                                  *
004365                                  *
004366                                  *
004367                                  *  M E S S A G E S
004368                                  *
004369                                        IFZ   $AF-2,LAF1
004370  1296  6370 6673 3420              MSG1  TEXT  'CPFS4   WCS TEST'
        1299  2020 7763 7320
              7465 7374
004371                                  LAF1  NULL
004374                                  SAF1  NULL
004375  129E  2020 2041 5052                    TEXT  '   APR 1978   REV. A$'
        12A1  2031 3937 3820
              2020 5245 562E
              2041 2400
004376  12A9  6D6F 6465 2400              MSG1A TEXT  'MODE$'
004377  12AC  7365 7175 656E              MSG1B TEXT  'SEQUENTIAL MODE - HEX SWITCH SHOULD BE "F"$'
        12AF  7469 616C 206D
              6F64 6520 2D20
              6865 7820 7377
              6974 6368 2073
              686F 756C 6420
              6265 2022 6622
              2400
004378  12C2  7472 616E 7370              MSG1C TEXT  'TRANSPARENT MODE - HEX SWITCH SHOULD BE "0"$'
        12C5  6172 656E 7420
              6D6F 6465 202D
              2068 6578 2073
              7769 7463 6820
              7368 6F75 6C64
              2062 6520 2230
              2224
004379  12D8  7468 6520 3173              MSG1D TEXT  'THE 1ST ERROR MESSAGE SHOULD ALWAYS BE',Z'0D0A';
        12DB  7420 6572 726F
004380  12DE  7220 6D65 7373                    '  CONSIDERED 1ST.  SUBSEQUENT ERROR$'
        12E1  6167 6520 7368
              6F75 6C64 2061
              6C77 6179 7320
              6265 0D0A 2020
              636F 6E73 6964
              6572 6564 2031
              7374 2E20 2073
              7562 7365 7175
              656E 7420 6572
              726F 7224
004381  12FE  2020 6D65 7373              MSG1E TEXT  '  MESSAGES COULD BE CAUSED BY PREVIOUS ERRORS$'
        1301  6167 6573 2063
              6F75 6C64 2062
              6520 6361 7573
              6564 2062 7920
              7072 6576 696F
              7573 2065 7272
              6F72 7324
004382  1315  6E61 6B24                  MSG2  TEXT  'NAK$'
004383  1317  6163 6B24                  MSG3  TEXT  'ACK$'
004384  1319  2020 202A 2072              MSG4  TEXT  '   * REPLACE MB *$'
        131C  6570 6C61 6365
              206D 6220 2A24
004385  1322  2020 202A 2072              MSG4A TEXT  '   * REPLACE DB 1 *$'
        1325  6570 6C61 6365
              2064 6220 3120
              2A24
004386  132C  2020 202A 2072              MSG4B TEXT  '   * REPLACE DB 2 *$'
        132F  6570 6C61 6365
              2064 6220 3220
              2A24
004387  1336  2020 202A 206D              MSG4C TEXT  '   * MAY BE WCS MB *$'
        1339  6179 2062 6520
              7763 7320 6D62
              202A 2400
004388  1341  2020 202A 2065              MSG4D TEXT  '   * EITHER WCS MB OR DB *$'
        1344  6974 6865 7220
              7763 7320 6D62
              206F 7220 6462
              202A 2400
004389  134F  2020 6966 2065              MSG4E TEXT  '  IF ERROR RESTRICTED TO ONE BIT OF ONE ZONE IN MAT$'
        1352  7272 6F72 2072
              6573 7472 6963
              7465 6420 746F
              206F 6E65 2062
              6974 206F 6620
              6F6E 6520 7A6F
              6E65 2069 6E20
              6D61 7424
004390  1369  2020 6572 726F              MSG4F TEXT  '  ERRORS FOLLOWING, MOST LIKELY BAD MEMORY CHIP ON DB.$'
        136C  7273 2066 6F6C
              6C6F 7769 6E67
              2C20 6D6F 7374
              206C 696B 656C
              7920 6261 6420
              6D65 6D6F 7279
              2063 6869 7020
              6F6E 2064 622E
              2400
004391  1385  7374 6174 7573              MSG5  TEXT  'STATUS$'
        1388  2400
004392  1389  4F2E 4B2E 2400              MSGOK TEXT  'O.K.$'
004393  138C  7365 7175 656E              MSGSP TEXT  'SEQUENTIAL MODE TEST NEXT,',Z'0D0A';
        138F  7469 616C 206D
004394  1392  6F64 6520 7465                    '  IF "O.K." NOT PRINTED IN 5 SEC, $'
        1395  7374 206E 6578
              742C 0D0A 2020
              6966 2022 4F2E
              4B2E 2220 6E6F
              7420 7072 696E
              7465 6420 696E
              2035 2073 6563
```

```
                2C20 2400
004395  13AC    2020 7468 656E      MSGSQ   TEXT  '  THEN MB FAULT.  THIS MESSAGE WILL',Z'0D0A';
        13AF    204D 4220 6661
004396  13B2    756C 742E 2020              '  NOT BE PRINTED IN SUBSEQUENT PASSES!$'
        13B5    7468 6973 206D
                6573 7361 6765
                2077 696C 6C0D
                0A20 206E 6F74
                2062 6520 7072
                696E 7465 6420
                696E 2073 7562
                7365 7175 656E
                7420 7061 7373
                6573 2124
004397  13D2    7472 616E 7370      MSGSR   TEXT  'TRANSPARENT MODE TEST NEXT,',Z'0D0A';
        13D5    6172 656E 7420
004398  13D8    6D6F 6465 2074              '  IF "O.K." NOT PRINTED IN 5 SEC,$'
        13DB    6573 7420 6E65
                7874 2C0D 0A20
                2069 6620 226F
                2E6B 2E22 206E
                6F74 2070 7269
                6E74 6564 2069
                6E20 3520 7365
                632C 2400
004399  13F2    6170 7431 2400      MAPT1   TEXT  'APT1$'
004400  13F5    6170 7432 2400      MAPT2   TEXT  'APT2$'
004401  13F8    6170 7433 2400      MAPT3   TEXT  'APT3$'
004402  13FB    6170 7434 2400      MAPT4   TEXT  'APT4$'
004403  13FE    6170 7435 2400      MAPT5   TEXT  'APT5$'
004404  1401    6170 7436 2400      MAPT6   TEXT  'APT6$'
004405  1404    6470 7431 2400      MDPT1   TEXT  'DPT1$'
004406  1407    6470 7432 2400      MDPT2   TEXT  'DPT2$'
004407  140A    6470 7433 2400      MDPT3   TEXT  'DPT3$'
004408  140D    6470 7434 2400      MDPT4   TEXT  'DPT4$'
004409  1410    6D61 7424           MMAT    TEXT  'MAT$'
004410  1412    2020 2069 7324      MSG6    TEXT  '   IS$'
004411  1415    2C20 2073 6224      MSG7    TEXT  ',  SB$'
004412  1418    6964 2400           MSG8    TEXT  'ID$'
004413                              LF2     IFZ   $AF-2,LAF2
004414  141A    6370 6673 3420      MSG9    TEXT  'CPF54 PS$'
        141D    7073 2400
004415                              LAF2    NULL
004418                              SAF2    NULL
004419  141F    3320 6D69 6E20      MSG10   TEXT  '3 MIN DONE$'
        1422    646F 6E65 2400
004420  1425    2020 2073 7461      MSG11   TEXT  '   STATUS =$'
        1428    7475 7320 3D24
004421  142B    7261 6D20 6E6F      MSG12   TEXT  'RAM NO.1 PRESENT$'
        142E    2E31 2070 7265
                7365 6E74 2400
004422  1434    7261 6D20 6E6F      MSG13   TEXT  'RAM NO.1 NOT PRESENT$'
        1437    2E31 206E 6F74
                2070 7265 7365
                6E74 2400
004423  143F    7261 6D20 6E6F      MSG14   TEXT  'RAM NO.2 PRESENT$'
        1442    2E32 2070 7265
                7365 6E74 2400
004424  1448    7261 6D20 6E6F      MSG15   TEXT  'RAM NO.2 NOT PRESENT$'
        144B    2E32 206E 6F74
                2070 7265 7365
                6E74 2400
004425  1453    6461 7461 203D      MSG19   TEXT  'DATA =$'
        1456    2400
004426  1457    2C20 206C 6F63      MSG21   TEXT  ',  LOC$'
        145A    2400
004427  145B    2C20 207A 6F6E      MSG22   TEXT  ',  ZONE$'
        145E    6524
004428  145F    2C20 2073 7461      MSG23   TEXT  ',  STATUS =$'
        1462    7475 7320 3D24
004429  1465    2C20 2020 7465      MSG24   TEXT  ',   TEST NO. =$'
        1468    7374 206E 6F2E
                203D 2400
004430                              *
004431  146D    7372 3124           MSGSR1  TEXT  'SR1$'
004432  146F    6163 3124           MSGAC1  TEXT  'AC1$'
004433  1471    7261 3124           MSGRA1  TEXT  'RA1$'
004434  1473    7374 3124           MSGST1  TEXT  'ST1$'
004435  1475    6362 3124           MSGCB1  TEXT  'CB1$'
004436  1477    6373 3124           MSGCS1  TEXT  'CS1$'
004437  1479    6372 3124           MSGCR1  TEXT  'CR1$'
004438  147B    6C62 3124           MSGLB1  TEXT  'LB1$'
004439  147D    7A6E 3124           MSGZN1  TEXT  'ZN1$'
004440  147F    6966 3124           MSGIF1  TEXT  'IF1$'
004441  1481    7469 3124           MSGTI1  TEXT  'TI1$'
004442  1483    7361 3124           MSGSA1  TEXT  'SA1$'
004443  1485    7361 3224           MSGSA2  TEXT  'SA2$'
004444                              *
004445                              *
004446                              *
004447                              *
004448                              *  R E S E R V E D   A R E A
004449                              *
004450                              *
004451  1487    0000                TMP1    RESV  1,0               TEMPORARY STORE
004452  1488    0000                TMP2    RESV  1,0               TEMPORARY STORE
004453  1489    0000                TMP3    RESV  1,0               TEMPORARY STORE
004454  148A    0000                TMP4    RESV  2,0               TEMPORARY STORE
004455  148C    0000                STOR1   RESV  1,0               TEMPORARY STORE
004456                              *
004457  148D    0000                MODBIT  RESV  1,0               =HEX 0800 (BIT 4) IF SEQUENTIAL MODE
004458                              *                               =HEX 0040 (BIT 9) IF TRANSPARENT MODE
004459                              *
004460  148E    0000                STABIT  RESV  1,0               =HEX 0900 IF SEQUENTIAL MODE
004461                              *                               =HEX 0140 IF TRANSPARENT MODE
004462                              *
004463  148F    0000                PASCNT  RESV  $AF,0             PASS COUNTER
004464                              *
004465  1490    0000                B4DCML  RESV  $AF,0             FOR B4 (PASS COUNT) RESTORATION
004466                              *
004467  1491    0000                CPCHAN  RESV  1,0               *** ASSUME 0 FOR NOW ***
004468                              *
004469  1492    2610                ID      DC    Z'2610'           WCS ID
```

```
004470                                  *
004471                                  *
004472  1493  0000                      RED     RESV  1,0               RED ERROR FORCED VIA MAIN MEMORY
004473                                  *                               SWITCH STORED HERE
004474                                  *
004475  1494  0000                      DNRERF  RESV  1,0               ERROR FLAG FOR TEST DNR
004476                                  *
004477  1495  0000                      RAM1FL  RESV  1,0               =1 IF RAM NO.1 PRESENT, =0 IF NOT.
004478                                  *
004479  1496  0000                      RAM2FL  RESV  1,0               =1 IF RAM NO.2 PRESENT, =0 IF NOT.
004480                                  *
004481  1497  0000                      RAMADR  RESV  1,0               = HEX 0800 IF RAM 1 PRESENT.
004482                                  *                               = HEX 0C00 IF ONLY RAM 2 PRESENT.
004483                                  *
004484  1498  0000                      APTEF1  RESV  1,0               ERROR FLAG 1 FOR "APT"
004485                                  *
004486  1499  0000                      APTEF2  RESV  1,0               ERROR FLAG 2 FOR "APT"
004487                                  *
004488  149A  0000                      DPTEF1  RESV  1,0               ERROR FLAG FOR "DPT"
004489                                  *
004490  149B  0000                      SPHADR  RESV  1,0               SPLASH ADDRESS
004491                                  *
004492  149C  0000                      SPHCOD  RESV  1,0               SPLASH MICROINSTRUCTION CODE
004493                                  *
004494  149D  0000                      RTFLG   RESV  1,0
004495                                  *
004496  149E  0000                      WCSIZE  RESV  1,0               WCS LAST LOCATION ADDRESS
004497                                  *                               (0BFF FOR RAM1, 0FFF FOR RAM2)
004498                                  *
004499  149F  0000                      TRPFLG  RESV  1,0               TRAP FLAG
004500                                  *                               = 0 IF TRAPS, = 1 IF NOT
004501                                  *
004502  14A0  0000                      TSTCNT  RESV  1,0               TEST COUNTER TO STORE TEST NO. FOR DEBUG
004503                                  *
004504                                  *
004505                                  *
004506                                  * WCS CHANNEL NO., FUNCTION CODE TABLE
004507                                  *
004508                                  *  (SHOWN BELOW FOR CP CH. NO. 0.  IF NOT CP 0, BITS 8 AND 9 MODIFIED)
004509                                  *  NOTE: BITS 8 AND 9 ARE THE COMPLEMENT OF THE CP CH. NO..
004510                                  *
004511                                  *
004512  14A1                            WCSTBL  RESV  0
004513  14A1  03C1                      OUTCON  DC    Z'03C1'           OUTPUT CONTROL WORD
004514  14A2  03C5                      OUTADR  DC    Z'03C5'           OUTPUT ADDRESS
004515  14A3  03CF                      OUTDAT  DC    Z'03CF'           OUTPUT DATA
004516  14A4  03C7                      OUTTSK  DC    Z'03C7'           OUTPUT TASK WORD
004517  14A5  03CE                      INDAT   DC    Z'03CE'           INPUT DATA
004518  14A6  03D8                      INSTAT  DC    Z'03D8'           INPUT STATUS WORD
004519  14A7  03E6                      INID    DC    Z'03E6'           INPUT I.D. CODE
004520                                  *
004521                                  *
004522                                  *  EXIT FIRMWARE TO RETURN FROM WCS TO NATIVE FIRMWARE IN SEQUENTIAL MODE
004523                                  *
004524                                  *
004525  14A8  C023                      BUFF1   DC    Z'C023'           RETURN ROUTINE WHEN ONLY RAM1 PRESENT
004526  14A9  7E90                              DC    Z'7E90'
004527  14AA  2500                              DC    Z'2500'
004528  14AB  03FD                              DC    Z'03FD'
004529                                  *
004530  14AC  8026                              DC    Z'8026'
004531  14AD  DCD0                              DC    Z'DCD0'
004532  14AE  8000                              DC    Z'8000'
004533  14AF  03FE                              DC    Z'03FE'
004534                                  *
004535  14B0  00A3                              DC    Z'00A3'
004536  14B1  CB00                              DC    Z'CB00'
004537  14B2  B035                              DC    Z'B035'
004538  14B3  A3FF                              DC    Z'A3FF'
004539                                  *
004540  14B4  8013                              DC    Z'8013'
004541  14B5  C641                              DC    Z'C641'
004542  14B6  7000                              DC    Z'7000'
004543  14B7  8020                              DC    Z'8020'
004544                                  *
004545                                  *
004546                                  *
004547  14B8  C023                      BUFF2   DC    Z'C023'           RETURN ROUTINE WHEN RAM2 IS PRESENT
004548  14B9  7E90                              DC    Z'7E90'
004549  14BA  2500                              DC    Z'2500'
004550  14BB  07FD                              DC    Z'07FD'
004551                                  *
004552  14BC  8026                              DC    Z'8026'
004553  14BD  DCD0                              DC    Z'DCD0'
004554  14BE  8000                              DC    Z'8000'
004555  14BF  07FE                              DC    Z'07FE'
004556                                  *
004557  14C0  00A3                              DC    Z'00A3'
004558  14C1  CB00                              DC    Z'CB00'
004559  14C2  B035                              DC    Z'B035'
004560  14C3  A7FF                              DC    Z'A7FF'
004561                                  *
004562  14C4  8013                              DC    Z'8013'
004563  14C5  C641                              DC    Z'C641'
004564  14C6  7000                              DC    Z'7000'
004565  14C7  8020                              DC    Z'8020'
004566                                  *
004567                                  *
004568                                  *
004569                                  *  EXIT FIRMWARE TO RETURN TO NATIVE FIRMWARE IN TRANSPARENT MODE
004570                                  *
004571                                  *
004572  14C8  C023                      BUFF3   DC    Z'C023'           FOR ONLY RAM1 PRESENT
004573  14C9  7E90                              DC    Z'7E90'
004574  14CA  2500                              DC    Z'2500'
004575  14CB  03FD                              DC    Z'03FD'
004576                                  *
004577  14CC  8026                              DC    Z'8026'
004578  14CD  DCD0                              DC    Z'DCD0'
004579  14CE  8000                              DC    Z'8000'
004580  14CF  03FE                              DC    Z'03FE'
004581                                  *
004582  14D0  00A3                              DC    Z'00A3'
```

```
004583  14D1  CF00                        DC    Z'CF00'
004584  14D2  B000                        DC    Z'B000'
004585  14D3  03FF                        DC    Z'03FF'
004586                            *
004587  14D4  8893                        DC    Z'8893'
004588  14D5  B701                        DC    Z'B701'
004589  14D6  7000                        DC    Z'7000'
004590  14D7  0020                        DC    Z'0020'
004591                            *
004592                            *
004593                            *
004594  14D8  C023                BUFF4   DC    Z'C023'          FOR RAM2 PRESENT
004595  14D9  7E90                        DC    Z'7E90'
004596  14DA  2500                        DC    Z'2500'
004597  14DB  07FD                        DC    Z'07FD'
004598                            *
004599  14DC  8026                        DC    Z'8026'
004600  14DD  DCD0                        DC    Z'DCD0'
004601  14DE  8000                        DC    Z'8000'
004602  14DF  07FE                        DC    Z'07FE'
004603                            *
004604  14E0  00A3                        DC    Z'00A3'
004605  14E1  CF00                        DC    Z'CF00'
004606  14E2  B000                        DC    Z'B000'
004607  14E3  07FF                        DC    Z'07FF'
004608                            *
004609  14E4  8893                        DC    Z'8893'
004610  14E5  B701                        DC    Z'B701'
004611  14E6  7000                        DC    Z'7000'
004612  14E7  0020                        DC    Z'0020'
004613                            *
004614                            *
004615                            *
004616                            *
004617                            *
004618                            *  D E B U G   L O O P
004619                            *
004620  14E8  9800 14A2                   LDR   $R1,<OUTADR
004621  14EA  A800 14A3                   LDR   $R2,<OUTDAT
004622  14EC  B800 14A5                   LDR   $R3,<INDAT
004623  14EE  C870 5555                   LDR   $R4,=Z'5555'     DATA WRITTEN
004624  14F0  8755                        CL    =$R5             INDICATOR
004625  14F1  8756                        CL    =$R6             DATA READ
004626  14F2  F870 0800                   LDR   $R7,=Z'0800'     ADRS OF 1ST LOCATION OF RAM 1
004627                            *
004628  14F4  0000                        HLT                    ALTER REGISTERS IF WANT.
004629                            *                              OTHERWISE HIT EXECUTE.
004630                            *
004631  14F5  8057                DEBUG   IO    =$R7,=$R1        OUTPUT ADRS
      14F6  0051
004632  14F7  8054                        IO    =$R4,=$R2        OUTPUT DATA
      14F8  0052
004633  14F9  8054                        IO    =$R4,=$R2        OUTPUT DATA
      14FA  0052
004634  14FB  8054                        IO    =$R4,=$R2        OUTPUT DATA
      14FC  0052
004635  14FD  8054                        IO    =$R4,=$R2        OUTPUT DATA
      14FE  0052
004636  14FF  0F7F                        NOP   >$-1
004637  1500  0F7F                        NOP   >$-1
004638  1501  0F7F                        NOP   >$-1
004639  1502  0F7F                        NOP   >$-1
004640  1503  0F7F                        NOP   >$-1
004641  1504  0F7F                        NOP   >$-1
004642  1505  0F7F                        NOP   >$-1
004643  1506  0F7F                        NOP   >$-1
004644  1507  8057                        IO    =$R7,=$R1        OUTPUT ADRS
      1508  0051
004645  1509  8056                        IO    =$R6,=$R3        INPUT DATA
      150A  0053
004646  150B  8056                        IO    =$R6,=$R3        INPUT DATA
      150C  0053
004647  150D  8056                        IO    =$R6,=$R3        INPUT DATA
      150E  0053
004648  150F  8056                        IO    =$R6,=$R3        INPUT DATA
      1510  0053
004649  1511  0F7F                        NOP   >$-1
004650  1512  0F7F                        NOP   >$-1
004651  1513  0F7F                        NOP   >$-1
004652  1514  0F7F                        NOP   >$-1
004653  1515  0F7F                        NOP   >$-1
004654  1516  0F7F                        NOP   >$-1
004655  1517  0F7F                        NOP   >$-1
004656  1518  0F7F                        NOP   >$-1
004657  1519  8654                        CPL   =$R4             COMPLEMENT DATA
004658  151A  8AD5                        INC   =$R5
004659  151B  5B00 14F5                   BEVN  $R5,<DEBUG       LOOP
004660                            *
004661  151D  8070 0000                   IO    =Z'0000',<OUTCON OUTPUT SIGNAL FOR SCOPE
      151F  0000 14A1
004662  1521  0F80 14F5                   B     <DEBUG           LOOP
004663                            *
004664  1523  434F 5059 5249              TEXT  'COPYRIGHT    1978, BY HONEYWELL INC.$'
      1526  4748 5420 2020
            2031 3937 382C
            2042 5920 484F
            4E45 5957 454C
            4C20 494E 432E
            2400
004665  1536  0100                        END   CPFS4,START
0000  ERR COUNT
      $AF    325C   390    406    773    806    880    919   2343C  2344C  2347C
             2357  2427   3666C  3757   3977   3982   4067C  4071   4078C  4092
             4102  4118C  4119C  4122C  4126   4153C  4154C  4157C  4161   4191C
             4192C 4195C  4200   4286   4345C  4350   4353C  4369   4372   4413
             4416  4463   4465
      $B1     338   343   1258   1271   1280   1973   2052C  2053   2171C  2172
             2303C 2304
      $B2    2455  2465   2470   2503   2590   2880   2881   2882   2883   2911
             2917  2921   2928   2971   2989   3006   3019   3023   3028   3031
             3061  3067   3074   3118   3136   3151   3166   3171   3174   3203
             3209  3216   3253   3270   3282   3296   3301   3304   3307   3334
             3338  3342   3352   3400   3446   3477   3478C  3530   3531C  3536
```

```
                3578    3626    3632    3636    3643    3673    3677    3770    3776    3793
                3826    3857    3887    3917    3922    3927    3932    3961    3965    3982
                3998    4028    4053C
$B3             3369    3402    4295    4300C   4312    4323    4345C   4350    4353C
$B4              348    4079
$B5            2355B   4124B   4159B   4198B
$B6             377B    383B    402B    413B    449B    455B    463B    470B    478B    486B
                494B    501B    508B    518B    526B    540B    548B    556B    566B    587B
                593B    600B    607B    615B    623B    632B    639B    646B    656B    664B
                678B    686B    695B    705B    731B    738B    748B    756B    765B    783B
                791B    802B    819B    830B    839B    849B    859B    870B    890B    900B
                913B    940B    948B    954B    960B    966B    972B    979B    989B    999B
               1017B   1025B   1031B   1037B   1043B   1050B   1057B   1067B   1077B   1105B
               1110B   1118B   1125B   1134B   1139B   1146B   1154B   1160B   1166B   1173B
               1178B   1186B   1193B   1201B   1206B   1211B   1218B   1231B   1276B   1286B
               1335B   1361B   1368B   1373B   1378B   1384B   1389B   1394B   1404B   1409B
               1415B   1425B   1432B   1437B   1443B   1455B   1460B   1465B   1471B   1476B
               1481B   1487B   1494B   1504B   1509B   1513B   1519B   1529B   1538B   1548B
               1647B   1672B   1695B   1698B   1700B   1704B   1845B   1857B   1881B   1890B
               1909B   1916B   1961B   2005B   2018B   2032B   2047B   2080B   2091B   2101B
               2114B   2124B   2139B   2149B   2166B   2206B   2223B   2236B   2251B   2263B
               2276B   2287B   2298B   2322B   2341C   2387B   2429B   2449B   2456B   2457B
               2463B   2473B   2483B   2501B   2559B   2591B   2592B   2598B   2603B   2654B
               2685B   2694B   2703B   2884B   2690B   2912B   2913B   2918B   2919B   2922B
               2923B   2929B   2930B   2942B   2972B   2973B   2990B   2991B   3007B   3008B
               3020B   3021B   3024B   3025B   3062B   3063B   3068B   3069B   3075B   3076B
               3087B   3119B   3120B   3137B   3138B   3152B   3153B   3167B   3168B   3204B
               3205B   3210B   3211B   3219B   3220B   3230B   3256B   3257B   3271B   3272B
               3285B   3286B   3297B   3298B   3335B   3336B   3339B   3340B   3343B   3344B
               3353B   3354B   3360B   3376B   3401B   3403B   3447B   3449B   3471B   3473B
               3483B   3545B   3546B   3549B   3555B   3561B   3627B   3628B   3633B   3634B
               3637B   3638B   3644B   3645B   3648B   3657B   3674B   3675B   3678B   3679B
               3759B   3794B   3795B   3796B   3806B   3827B   3828B   3829B   3839B   3858B
               3859B   3860B   3870B   3888B   3889B   3890B   3899B   3918B   3919B   3923B
               3924B   3928B   3929B   3933B   3934B   3935B   3944B   3962B   3963B   3966B
               3967B   3968B   3985B   4054B   4117C   4141B   4152C   4176B   4190C   4215B
               4225B   4234B   4243B   4252B   4267B   4282B   4303B   4321B
$B7            3660    3666C   3977    3993C   3998
$R1             340     341     342     343     352     353C    354C    355     356C    515
                522B    653     660B    771     772C    813     814C    878     879C    976
                985B   1054    1063B   1096    1098    1099C   1261    1262    1308    1317
               1320    1321B   1323    1324    1325B   1599    1600C   1614    1615C   1736
               1761    1766    1779    1783    1795    1798    1809    1811    1932    1933C
               1938    1939    1940B   1942    1943C   1944    1947    1948C   1949    1955B
               1966    1996    1998    2010    2023    2037    2073    2082    2084    2103
               2106    2126    2130    2151    2156    2196    2208    2213    2227    2238
               2242    2255    2265    2268    2280    2289    2291    2396    2398    2400
               2401    2438    2439C   2440    2441C   2453    2458B   2466    2467C   2486C
               2491    2504    2505C   2518    2519C   2520    2521C   2522C   2525    2576
               2577C   2578    2579C   2589    2593    2594    2609    2616    2643    2644
               2653    2699    2700C   2706C   2710    2938    2939C   2944C   2962    2966
               2967    2968    2978    2997    3001    3002    3003    3014    3083    3084C
               3089C   3107    3113    3114    3115    3125    3142    3147    3148    3158
               3226    3227C   3232C   3247    3262    3276    3291    3365    3366C   3374
               3406    3443    3450    3451    3455    3456    3533    3547B   3562C   3570
               3653    3654C   3659C   3664    3762    3763C   3764    3765C   3771    3772C
               3773    3774    3800    3801C   3833    3834C   3864    3865C   3894    3895C
               3939    3940C   3972    3973C   3974    3976C   3978    3979C   3980    3981C
               3990    3991    3994    3995    4029    4030C   4031    4052    4053C   4084
               4085    4089    4090C   4116C   4120    4121B   4129    4151C   4155    4156B
               4164    4189C   4193    4194B   4196    4620    4631    4644
$R2             339     344B   1170    1182B   1269    1290    1310B   1364    1398B   1737
               1765    1782    1797    1810    1952C   1954    1967    1997    2003    2009
               2016    2022    2030    2036    2045    2078    2083    2089    2099    2105
               2112    2122    2129    2137    2147    2155    2164    2204    2212    2221
               2234    2241    2249    2261    2267    2274    2285    2290    2296    2377C
               2380    2382C   2385    2397    2398    2399C   2478    2491    2493C   2495
               2523    2526B   2541C   2542    2543    2547C   2556    2583    2585    2594
               2614    2615    2616    2712    2715    2726    2945C   3090C   3109    3127
               3144    3160    3233C   3346    3347    3348C   3355    3356    3377    3382
               3479    3480C   3539    3540    3553    3554C   3572C   3597    3598    3775
               3777B   4072    4077B   4621    4632    4633    4634    4635
$R3            1224    1234    1256    1274    1293C   1296    1298C   1303    1658    1663
               1738    1762    1763    1764    1767    1768    1769    1770    1780    1781
               1784    1785    1786    1796    1799    1800    1812    1953    1954    1968
               1999    2004    2011    2012    2017    2024    2025    2026    2031    2038
               2039    2040    2041    2046    2074    2079    2085    2090    2095    2100
               2104    2107    2108    2113    2118    2123    2127    2128    2131    2132
               2133    2138    2143    2148    2152    2153    2154    2157    2158    2159
               2160    2165    2197    2198    2199    2200    2205    2209    2210    2211
               2214    2215    2216    2217    2222    2228    2229    2230    2235    2239
               2240    2243    2244    2245    2250    2256    2257    2262    2266    2269
               2270    2275    2281    2286    2292    2297    2342C   2345    2346B   2351
               2353    2360B   2386    2479    2480C   2494C   2607    2608C   2925    2926
               2927    2951    2952    2954    2956    3071    3072    3073    3097    3098
               3100    3102    3213    3214    3215    3238    3239    3241    3243    3250
               3251    3252    3279    3280    3281    3349    3350    3356    3371    3372
               3382    3390    3444    3445    3451    3459    3463    3464    3495    3532
               3573B   3640    3641    3642    3802    3835    3866    3941    3949    4291C
               4300C   4301    4311C   4326C   4327    4346    4347    4348C   4349    4354
               4355    4356C   4357    4622    4645    4646    4647    4648
$R4             381     387     389C    395     405C    411     419     427C    453     460
                476     484     492     499     506     524     531     538     546     554
                563     591     598     605     613     621     630     637     654     662
                669     676     684     693     702     746     752     763     781     787
                798     836     842     847     855     868     888     896     909     952
                964     977     987     996    1023    1035    1047    1065    1074    1116
               1123    1137    1144    1152    1158    1184    1191    1209    1216    1226
               1229    1259    1260C   1266    1267C   1271    1278C   1279    1280    1281
               1290    1291    1292C   1297B   1330    1333    1407    1413    1423    1430
               1435    1441    1485    1492    1524    1527    1542    1545    1589    1590
               1606    1607    1632    1633    1636C   1637    1638    1659C   1660    1661
               1756    1765    1771    1773    1782    1787    1789    1797    1801    1803
               1810    1813    1815    1820    1841    1842    1850C   1851    1852    1862C
               1863C   1864    1865    1867    1957    1959    1969    1997    2000    2004
               2009    2013    2017    2022    2027    2031    2036    2042    2046    2075
               2079    2096    2100    2119    2123    2144    2148    2212    2218    2222
               2241    2246    2250    2267    2271    2275    2290    2293    2297    2310
               2317    2318    2335C   2336    2337    2339    2442    2648    2660    2665
               2677    2687    2880    2886    3375    3377    3389    3390    3398C   3404
               3467    4205C   4261C   4262C   4263    4264C   4266    4294    4301    4304
               4323    4324C   4623    4632    4633    4634    4635    4657C
$R5             459     460     482     483     484     498     499     544     545     546
```

```
                562     563     597     598     619     620     621     636     637     682
                683     684     701     702     754     762     763     789     796     797
                798     800     857     867     868     898     907     908     909     911
                994     995     996    1072    1073    1074    1121    1122    1123    1142
               1143    1144    1157    1158    1189    1190    1191    1214    1215    1216
               1227    1228    1229    1254    1255    1281    1331    1332    1333    1338
               1412    1413    1428    1429    1430    1440    1441    1490    1491    1492
               1525    1526    1527    1543    1544    1545    1703    1760    1774B   1778
               1790B   1794    1804B   1808    1816B   1844    1958    1959    1970    2083
               2086    2090    2105    2109    2113    2129    2134    2138    2155    2161
               2165    2201    2205    2231    2235    2258    2262    2282    2286    2311
               2321    2443    2549C   2550    2555    2649    2661    2666    2678    2688
               2881    2887    3373    3384    3385    3407    3468    4037    4075    4076
               4078C   4207    4211C   4322    4329B   4624C   4658C   4659B
        $R6    1204    1257    1262    1279    1308    1324    1582    1584    1590    1607
               1627    1633    1663    1753    1754C   1762    1763    1764    1767    1768
               1769    1770    1771    1780    1781    1784    1785    1786    1787    1796
               1799    1800    1801    1812    1813    1895    1896C   1984    1985C   1999
               2000    2011    2012    2013    2024    2025    2026    2027    2038    2039
               2040    2041    2042    2060    2061C   2074    2075    2085    2086    2095
               2096    2104    2107    2108    2109    2118    2119    2127    2128    2131
               2132    2133    2134    2143    2144    2152    2153    2154    2157    2158
               2159    2160    2161    2178    2179B   2183    2184C   2197    2198    2199
               2200    2201    2209    2210    2211    2214    2215    2216    2217    2218
               2228    2229    2230    2231    2239    2240    2243    2244    2245    2246
               2256    2257    2258    2266    2269    2270    2271    2281    2282    2292
               2293    2366C   2444    2469    2471B   2650    2662    2667    2679    2689
               2882    2888    3395    3396    3397    3469    4073C   4075    4298    4300C
               4318    4327    4333    4625C   4645    4646    4647    4648
        $R7     446C    468     516     583C    644     722C    735     741     946     958
                970    1029    1041    1055    1253C   1274    1301C   1307C   1574C   1576
               1578    1620    1622    1751    1761    1766    1779    1783    1795    1798
               1809    1811    1826    1834    1869    1983    1996    1998    2010    2023
               2037    2051C   2059    2073    2082    2084    2103    2106    2126    2130
               2151    2156    2170C   2180    2182    2196    2208    2213    2227    2238
               2242    2255    2265    2268    2280    2289    2291    2302C   2349    2373C
               2445    2448    2450    2462    2464    2468    2500    2502    2651    2657
               2658C   2669    2670    2671C   2673    2674    2675    2683    2692    2720
               2723    2883    2889    3402    3448C   3458    3465    3493    3495    3542
               3803    3811    3836    3844    3867    3875    3896    3904    4071    4074
               4626    4631    4644
2622   AC1     2620
4159   ACKEK1  4156B
4161   ACKEK2  4152C   4153C   4154C   4157C
4164   ACKEK3  4160B
4178   ACKEK4  4151C   4164    4167    4174
4151   ACKEKR   518B    656B    979B   1057B   1178B   1394B   1481B   1519B
2573   ACT10   2512B   2513B
2592   ACT20   2595B
2603   ACT30   2596B
2612   ACT40   2608C   2610B
2620   ACT50   2617B
4169   AKEK3A  4169
4173   AKEK3B  4168B
4176   AKEK3C  4170B
1564   API     1546B
1605   API1    1591B
1687   API10   1681B
1690   API11   1690
1695   API12   1678B   1685B
1698   API13   1653B   1693B
1700   API14   1648B   1673B   1696B
1703   API15   1639B   1662B
1619   API2    1601B   1608B
1643   API3    1643
1650   API4    1641B   1650
1657   API5    1634B
1669   API6    1669
1675   API7    1667B   1675
1680   API8    1664B
1683   API9    1683
4484   APIEF1  1569C   1600C   1640    1680
4486   APIEF2  1570C   1615C   1666    1687
4465   B4DCML  4078C   4079
4525   BUFF1   2465
4547   BUFF2   2503
4572   BUFF3   3770
4594   BUFF4   4028
3037   CB1     3035
2906   CBT10   2717B   2728B   2734B
3014   CBT100  2958B
3019   CBT150  2916B
3028   CBT160  2970B
3031   CBT170  3005B
2925   CBT20   3026B
3035   CBT200  2963B   2979B   2998B   3015B
2938   CBT30   2975B   2993B   3010B
2947   CBT40   2939C
2951   CBT45   3038B
2962   CBT50   2953B
2972   CBT60   3029B
2978   CBT70   2955B
2997   CBT80   2957B
3007   CBT90   3032B
       CONSID  4380
4467   CPCHAN   340
       CPFS4      1    4665
3313   CR1     3311
3198   CRT10   3162B
3291   CRT100  3245B
3296   CRT150  3208B
3301   CRT160  3218B
3304   CRT170  3255B
3307   CRT180  3284B
3213   CRT20   3299B
3311   CRT200  3248B   3263B   3277B   3292B
3219   CRT25   3302B
3226   CRT30   3259B   3274B   3288B
3235   CRT40   3227C
3238   CRT45   3314B
3247   CRT50   3240B
3256   CRT60   3305B
3262   CRT70   3242B
```

```
3276  CRT80    3244B
3285  CRT90    3308B
3182  CS1      3180
2880  CSB10    2456B  2591B  2912B  2918B  2922B  2929B  2972B  2990B  3007B  3020B
               3024B  3062B  3068B  3075B  3119B  3137B  3152B  3167B  3204B  3210B
               3219B  3256B  3271B  3285B  3297B  3335B  3339B  3343B  3353B  3401B
               3447B  3545B  3627B  3633B  3637B  3644B  3674B  3678B  3795B  3828B
               3859B  3889B  3918B  3923B  3928B  3934B  3962B  3967B
2886  CSB20    2449B  2457B  2463B  2501B  2592B  2654B  2685B  2694B  2913B  2919B
               2923B  2930B  2973B  2991B  3008B  3021B  3025B  3063B  3069B  3076B
               3120B  3138B  3153B  3168B  3205B  3211B  3220B  3257B  3272B  3286B
               3298B  3336B  3340B  3344B  3354B  3403B  3449B  3471B  3546B  3628B
               3634B  3638B  3645B  3675B  3679B  3796B  3829B  3860B  3890B  3919B
               3924B  3929B  3935B  3963B  3968B
3056  CST10    3016B
3158  CST100   3104B
3166  CST150   3066B
3171  CST160   3117B
3174  CST170   3150B
3071  CST20    3169B
3180  CST200   3108B  3110B  3126B  3128B  3143B  3145B  3159B  3161B
3083  CST30    3122B  3140B  3155B
3092  CST40    3084C
3097  CST45    3183B
3107  CST50    3099B
3119  CST60    3172B
3125  CST70    3101B
3142  CST80    3103B
3152  CST90    3175B
4631  DEBUG    4659B  4662B
4073  DIGIT    4077B
 722  DNR       650B   699B   703B
 735  DNR1      730B
 613  DNR10     732B   739B   749B   757B   766B   784B   792B   803B
 828  DNR11     774B   777B   807B   810B   818B
 836  DNR12     829B
 842  DNR13     837B
 855  DNR14     848B
 867  DNR15     856B
 878  DNR16     869B
 888  DNR17     843B
 896  DNR18     889B
 907  DNR19     897B
 741  DNR2      736B
 919  DNR20     910B
 925  DNR21     881B   884B   920B
 946  DNR22     939B
 952  DNR23     947B
 958  DNR24     953B
 964  DNR25     959B
 970  DNR26     965B
 976  DNR27     971B
 977  DNR28     985B
 985  DNR29     978B
 752  DNR3      747B
 994  DNR30     988B
1008  DNR31     931B
1015  DNR32     933B
1023  DNR33    1016B
1029  DNR34    1024B
1035  DNR35    1030B
1041  DNR36    1036B
1047  DNR37    1042B
1054  DNR38    1048B
1055  DNR39    1063B
 762  DNR4      753B
1063  DNR40    1056B
1072  DNR41    1066B
 771  DNR5      764B
 781  DNR6      742B
 787  DNR7      782B
 796  DNR8      788B
 806  DNR9      799B
4475  DNRERF    723C   814C   925
 328  DONE     4097B
1736  DPT      1565B  1567B  1688B  1701B
1753  DPT1     1835B
1841  DPT10    1830B
1844  DPT11    1853B  1866B
1850  DPT12    1840B
1854  DPT13    1854
1760  DPT2     1823B
1761  DPT3     1774B
1779  DPT4     1790B
1795  DPT5     1804B
1809  DPT6     1816B
1829  DPT7     1747B
1834  DPT8     1897B
1839  DPT9     1827B
4488  DPTERF1  1745C  1839   1873   1896C
1878  DPTER1   1878
1886  DPTER2   1874B  1886
1893  DPTER3   1870B
1901  DPTER4   1872B  1894B  1901
1916  DPTER5   1858B  1882B  1891B
1862  DPTERR   1772B  1788B  1802B  1814B
4067  EOP      3670B  3687B  4020B  4042B
      ERED     4380
 338  ESTBL
 343  ESTBL1    344B
 411  GETID     391B   393B   407B
 419  GETID1    412B
 422  GETID2    422
4101  GLEN     4087B
4469  ID        430
      IF       4394   4398
3686  IF1      3684
3619  IFT10    3392B
3640  IFT20    3680B
3653  IFT30    3647B
3662  IFT40    3654C
3673  IFT60    3631B
3684  IFT80    3665B  3668B
```

```
4517  INDAT    506   605   654   836   842   952   964   977  1023  1035
              1047  1204  1296  1589  1606  1632  1658  1738  1968  2386
              4298  4622
4519  INID      411   524   662  1224  1317
 381  INIT1    376B
 387  INIT2    382B
 395  INIT3    388B
 397  INIT4    397
 405  INIT5    396B
4518  INSTAT    381   453   476   492   538   554   591   613   630   676
              693   746   781   847   888   987  1065  1116  1137  1152
             1184  1209  1226  1258  1330  1407  1423  1435  1485  1524
             1542  1637  1660  1841  1851  1864  1957  2317  2336  2380
             2542  4132  4167  4263
4371  LAF1     4369
4415  LAF2     4413
3424  LB1      3422
3329  LBT10   3293B
3354  LBT20   3357B
3365  LBT25   3359B
3376  LBT30   3391B
3379  LBT40   3366C
3389  LBT43   3408B
3395  LBT45   3386B
3411  LBT48    3369
3422  LBT50   3383B
 446  LET      417B  420B
 453  LET1     448B
 522  LET10    517B
 531  LET11    525B
 533  LET12     533
 538  LET13    527B  532B
 544  LET14    539B
 554  LET15    541B  547B
 562  LET16    555B
 459  LET2     454B
 468  LET3     456B  461B
 476  LET4     469B
 482  LET5     477B
 492  LET6     474B  479B  485B
 498  LET7     493B
 506  LET8     495B  500B
 515  LET9     507B
 516  LET9A    522B
4399  MAPT1    1643
4400  MAPT2    1650
4401  MAPT3    1669
4402  MAPT4    1675
4403  MAPT5    1683
4404  MAPT6    1690
1932  MAT     1843B 1846B 1912B
2073  MAT10   2173B
2082  MAT11   2076B
2095  MAT12   2087B
2103  MAT13   2097B
2118  MAT14   2110B
2126  MAT15   2120B
2143  MAT16   2135B
2151  MAT17   2145B
2170  MAT18   2162B
2182  MAT19   2179B
1947  MAT2    1940B
2183  MAT20   2181B
2196  MAT21   2305B
2208  MAT22   2202B
2227  MAT23   2219B
2238  MAT24   2232B
2255  MAT25   2247B
2265  MAT26   2259B
2280  MAT27   2272B
2289  MAT28   2283B
2302  MAT29   2294B
1951  MAT3    1945B
2321  MAT30   2338B
1954  MAT4    1955B
1983  MAT4A   2314B
1966  MAT4B   1960B
1996  MAT5    2054B
2009  MAT6    2001B
2022  MAT7    2014B
2036  MAT8    2028B
2051  MAT9    2043B
2349  MATER1  2346B
2353  MATER2  2350B
2355  MATER3  2352B
2357  MATER4  2341C 2343C 2344C 2347C
2360  MATER5  2356B
2363  MATER6  2360B
2365  MATER7  2362B
2389  MATERA  2366C  2367  2373C  2374  2377C  2378  2382C  2383
2390  MATERB  2335C  2339  2342C  2370
2335  MATERR  2005B 2018B 2032B 2047B 2080B 2091B 2101B 2114B 2124B 2139B
              2149B 2166B 2206B 2223B 2236B 2251B 2263B 2276B 2287B 2298B
2742  MC1      2911  3061  3203
2787  MC10     3067  3151
2792  MC11     3166  3174
2797  MC12     3118
2802  MC13     3171
2807  MC14     3074
2812  MC15     3209
2817  MC16     3296
2822  MC17     3253
2827  MC18     3304
2832  MC19     3216  3282
2747  MC2      2989  3136  3270
2837  MC20     3301  3307
2842  MC21     3338
2847  MC22     3342  3400
2852  MC23     2455  2590  3334  3446
2857  MC24     3626
2862  MC25     3632
2867  MC26     3673
2872  MC27     3643
```

```
2752  MC3       2917    3006
2757  MC4       3019    3031
2762  MC5       2971
2767  MC6       3028
2772  MC7       2921    3536    3636
2777  MC8       3023    3578    3677
2782  MC9       2928    3352
4405  MDPT1     1854
4406  MDPT2     187B
4407  MDPT3     1886
4408  MDPT4     1901
1359  MET       1321B
1364  MET1      1360B
1428  MET10     1424B
1435  MET11     1431B
1440  MET12     1436B
1453  MET13     1414B   1419B
1458  MET14     1454B
1463  MET15     1459B
1467  MET16     1464B
1474  MET17     1470B
1479  MET18     1475B
1485  MET19     1480B
1366  MET1A     1442B   1446B
1371  MET2      1367B
1490  MET20     1486B
1502  MET21     1493B
1507  MET22     1503B
1511  MET23     1508B
1516  MET24     1512B
1523  MET25     1518B
1535  MET26     1528B
1541  MET27     1537B
1376  MET3      1372B
1380  MET4      1377B
1387  MET4A     1383B
1392  MET5      1388B
1398  MET6      1393B
1407  MET7      1403B
1412  MET8      1408B
1423  MET9      1398B
4409  MMAT      2358
4457  MODBIT     389C    405C     459     483     498     545     562     597     620     636
              683     701     754     762     797     857     867     908     995    1073
             1122    1143    1157    1190    1215    1228    1255    1332    1338    1412
             1429    1440    1491    1526    1544    1638    1661    1703    1842    1844
             1852    1865    1958    2318    2321    2337    2397
2396  MODCK     23198   2323B
4370  MSG1       333
4373  MSG1
4419  MSG10     4093
4420  MSG11     4138    4173    4260
4421  MSG12      775
4422  MSG13      808
4423  MSG14      882
4424  MSG15      921
4425  MSG19     1289    2365
4376  MSG1A      397
4377  MSG1B      392
4378  MSG1C      408
4379  MSG1D      359
4381  MSG1E      360
4382  MSG2      4128    4134    4169
4426  MSG21     2372
4427  MSG22     2376
4428  MSG23     2381    2546
4429  MSG24     2551
4383  MSG3      4163
4384  MSG4       431     536     674    1239    2531    2621    2732    3036    3181    3312
             3423    3506    3587    3594    3685    4039    4131    4166    4213    4224
4385  MSG4A     2361    4233
4386  MSG4B     2363    4242
4387  MSG4C     4251
4388  MSG4D     1908
4389  MSG4E     1910
4390  MSG4F     1911
4391  MSG5      4202
4410  MSG6       426    4204
4411  MSG7       429    1300    2369    4210
4412  MSG8       422     533     671    1236
4414  MSG9      4091    4101
4417  MSG9
4432  MSGAC1    2620
4435  MSGCB1    3035
4437  MSGCR1    3311
4436  MSGCS1    3180
4440  MSGIF1    3684
4438  MSGLB1    3422
4392  MSGOK     3669    4017
4433  MSGRA1    2731
4442  MSGSA1    3586
4443  MSGSA2    3593
4393  MSGSP     2430
4395  MSGSQ     2431    3761
4397  MSGSR     3760
4431  MSGSR1    2530
4434  MSGST1    2545
4441  MSGT11    4038
4439  MSGZN1    3505
4124  NAKEK1    4121B
4126  NAKEK2    4117C   4118C   4119C   4122C
4129  NAKEK3    4125B
4143  NAKEK4    4116C   4129    4132    4139
4116  NAKEKR     377B    383B    413B    449B    455B    470B    478B    494B    508B    526B
              540B    556B    587B    593B    607B    615B    632B    646B    664B    678B
              695B    731B    738B    748B    783B    819B    830B    839B    849B    890B
              940B    948B    954B    960B    966B    972B    989B   1017B   1025B   1031B
             1037B   1043B   1050B   1067B   1105B   1110B   1118B   1134B   1139B   1154B
             1166B   1173B   1186B   1201B   1206B   1211B   1276B   1361B   1368B   1373B
             1378B   1384B   1389B   1404B   1409B   1425B   1437B   1455B   1460B   1465B
             1471B   1476B   1487B   1504B   1509B   1513B   1538B   3561B
 583  NERT       512B    560B    564B
 591  NERT1      586B
```

```
   3487  ZNT40     3480C
   3493  ZNT50
   3501  ZNT80     3477
   3505  ZNT90     3490B
         ZV$AF      307     2345    3990    4120    4155    4193
         ZV$ER      310      397B    422B    533B    671B   1236B   1643B   1650B   1669B   1675B
                   1683B   1690B   1854B   1878B   1886B   1901B   2355B   2530B   2545B   2620B
                   2731B   3035B   3180B   3311B   3422B   3505B   3586B   3593B   3684B   4038B
                   4124B   4134B   4159B   4169B   4198B
         ZV$LK      308     4295    4312
         ZV$RD      333B
         ZV$T       426B     429B    431B    536B    674B   1239B   1300B   1908B   2361B   2363B
                   2369B   2372B   2376B   2381B   2531B   2546B   2551B   2621B   2732B   3036B
                   3181B   3312B   3423B   3506B   3587B   3594B   3685B   4039B   4131B   4138B
                   4166B   4173B   4204B   4210B   4213B   4224B   4233B   4242B   4251B   4260B
         ZV$TC      359B     360B    392B    408B    775B    808B    882B    921B   1289B   1910B
                   1911B   2365B   2430B   2431B   3669B   3760B   3761B   4017B   4091B   4093B
                   4101B   4280B
         ZV$TD     2378B   4092B   4102B
         ZV$THZ     428B     430B   1299B   1302B   2367B   2370B   2374B   2383B   2548B   2552B
                   4139B   4174B   4206B   4212B   4265B   4325B
         ZV$TTY     309
 604 LABELS
3293 REFERENCES
4665 RECORDS
   4 U FLAGS
   2 M FLAGS
   8 N FLAGS
6 CROSS REF VERSION L - 24 SEPT, 1976
RS LINKER VERSION 5.00 04/19778 1832.5 EST WED
LINK MAP FOR CPFS4
 START          0100
 LOW            0000
 HIGH           1B4C
 CURRENT        1B4D
*LOC DEFS
  ZHCOMM        0000
 *CPFS4         0000  REV A
  ZHPFR         0000
  ZHTSA         0002
  ZHNTSA        0010
  ZHRTCI        0014
  ZHRTCC        0015
  ZHRTCL        0016
  ZHWDTC        0017
  ZHMERC        001F
  ZHIAFB        0020
  ZHTH29        0063
  ZHTH28        0064
  ZHTH27        0065
  ZHTH26        0066
  ZHTH25        0067
  ZHTH24        0068
  ZHTH23        0069
  ZHTH22        006A
  ZHTH21        006B
  ZHTH20        006C
  ZHTH19        006D
  ZHTH18        006E
  ZHTH17        006F
  ZHMEMP        006F
  ZHTH16        0070
  ZHLERR        0070
  ZHTH15        0071
  ZHNRES        0071
  ZHTH14        0072
  ZHPMEM        0072
  ZHTH13        0073
  ZHP-OP        0073
  ZHTH12        0074
  ZHTH11        0075
  ZHTH10        0076
  ZHTH9         0077
  ZHTH8         0078
  ZHTH7         0079
  ZHTH6         007A
  ZHOVFL        007A
  ZHTH5         007B
  ZHOP-N        007B
  ZHTH4         007C
  ZHTH3         007D
  ZHSC-N        007D
  ZHTH2         007E
  ZHTRC         007E
  ZHTH1         007F
  ZHMCL         007F
  ZHISAZ        0080
  ZHIVBS        0080
  ZHTVBS        0080
 *ZV$TH         1536
  ZV$THZ        155E
  ZV$TD         156B
  ZV$TH         1536
 *ZV$ER         1586  REV. 5.0
  ZV$ER         1586
  ZV$TA         15B2
  ZV$---0       1599
 *ZV$T          15F6  REV. 5.0
  ZV$TC         15FF
  ZV$T          15F6
  ZV$Q          1608
  ZV$QC         1613
 *ZV$QP         1627
  ZV$QP         1627
  ZV$---4       1647
 *ZV$HA         1653
  ZV$HA         1653
  ZV$HZ         165D
  ZV$HS         1658
 *ZV$HD         168C
  ZV$HD         168C
 *ZV$RD         16BE  REV. 7
  ZV$RD         16BE
```

```
  ZV$AF          16CF
  ZV$HR          16ED
  ZV$LR          16EA
  ZV$DAT         16CD
  ZV$SV1         1893
  ZV$TTY         16D1
  ZV$SV2         18A3
  ZV$OTP         1765
  ZV$IZ          16F8
  ZV$HM          1734
  ZV$HRU         16E7
  ZV$HRL         16E8
  ZV$LRU         16E9
  ZV$LRL         16EA
  ZV$HBD         16EB
  ZV$CF1         16D9
  ZV$CF2         16DA
  ZV$TK          16D6
  ZV$BKS         16E5
  ZV$BKF         16E6
  ZV$---5        16F0
  ZV$TID         16DD
  ZV$RMD         16CE
  ZV$BUD         16D2
  ZV$OLD         16DE
  ZV$NSR         16E3
  ZV$MCP         16EC
  HIBAUD         16EB
  ZV$RAK         16D7
  ZV$RAW         16D8
  ZV$ST1         16DB
  ZV$RDT         18EF
  ZV$CTL         16D5
  ZV$B1          1810
  ZV$RCC         16DC
  ZV$RCD         16DF
  ZV$STK         16E1
  ZV$TST         1945
  ZV$MDC         1919
  ZV$R99         1B17
  ZV$ISA         16F3
  ZV$UIH         16EE
  ZV$SV3         18B3
  ZV$ZRO         1772
  ZV$BSH         1774
  ZV$CPU         16D4
  ZV$R50         1752
  ZV$R60         175D
  ZV$RT          1A54
  ZV$ALL         16D3
 *MLCHPG         1B1C   T+V
  MLCHPG         1B1C
  ENDCHP         1B4D
*UNLINK MODULE(S)
 ZV$IC
 ZV$THZ
 ZV$T
```