```
000001                          TITLE DCMC1  'REV. A'
000002                  *
000003                  *
000004                  *
000005                  *
000006                  * DESCRIPTION:
000007                  *
000008                  *    THIS T&V PROGRAM VERIFIES PROPER OPERATION OF THE DUALLINE
000009                  *    COMMUNICATION PROCESSOR (DLCP) FOR SERIES 60, LEVEL 6/20.
000010                  *
000011                  *    THE SUBSYSTEM ITEMS SUPPORTED BY THIS PROGRAM ARE:
000012                  *
000013                  *       DLC9101 DUAL LINE COMMUNICATION PROCESSOR
000014                  *
000015                  *
000016                  * REVISION HISTORY:
000017                  *
000018                  *       REV      DATE
000019                  *        A       FEB 78    ORIGINAL RELEASE
000020                  *
000021                  *
000022                  ********************************************************************************
000023                  *    THIS  DOCUMENT  AND  THE  INFORMATION  CONTAINED
000024                  *    THEREIN IS CONFIDENTIAL AND PROPRIETARY TO AND THE
000025                  *    EXCLUSIVE  PROPERTY  OF  HONEYWELL INFORMATION
000026                  *    SYSTEMS INC.  IT  IS MADE AVAILABLE ONLY TO HONEY-
000027                  *    WELL AUTHORIZED RECIPIENTS FOR THEIR USE SOLELY IN
000028                  *    THE  MAINTENANCE  AND  OPERATION  OF  HONEYWELL
000029                  *    PRODUCTS.  THIS  DOCUMENT  AND INFORMATION MUST BE
000030                  *    MAINTAINED IN STRICTEST CONFIDENCE;  IT  MUST  NOT
000031                  *    BE  REPRODUCED  IN  WHOLE OR IN PART; AND IT SHALL
000032                  *    NOT BE DISCLOSED TO ANY OTHER  PARTY  WITHOUT  THE
000033                  *    PRIOR WRITTEN CONSENT OF HONEYWELL.
000034                  *
000035                  *
```

```
000036                          / PROGRAM PREPARATION:
000037                          *
000038                          *    THE ROOT SOURCE OF THIS PROGRAM, AFTER THE ADDITION OF APPROPRIATE
000039                          *    TITLE AND END STATEMENTS, WAS PROCESSED BY THE HOST RESIDENT ASSEMBLER
000040                          *    TO CREATE EITHER SHORT OR LONG ADDRESS FORM (SAF OR LAF) OBJECT TEXT
000041                          *    AND LISTING. THE OBJECT TEXT WAS FURTHER PROCESSED BY THE HOST RESIDENT
000042                          *    LINKER USING THE APPROPRIATE CONSOLE ZV$LIB LIBRARY TO CREATE A PUNCH
000043                          *    SEGMENT CONTAINING AN EXECUTABLE MODULE. THE ASSEMBLY LISTING WAS
000044                          *    AUGMENTED WITH CROSS REFERENCE DATA PLUS THE LOAD MAP FROM THE LINKER
000045                          *    TO CREATE A LIST SEGMENT.
000046                          *
000047                          *     DCMW1     60135054-001    (SOURCE)
000048                          *     DCMC1     60135055-001
000049                          *
000050                          * PROGRAM DISTRIBUTION:
000051                          *
000052                          *    THE ELEMENTARY ITEMS SUBMITTED TO THE I&V PROGRAM DISTRIBUTION CENTER
000053                          *    WERE THE EXECUTABLE PUNCHED CARD DECKS OF MLCS1 AND MLCL1, AND MAGNETIC
000054                          *    TAPE IMAGES OF THE AUGMENTED LISTINGS.
000055                          *
000056                          *    REPRODUCTIONS OF THE EXECUTABLE CARD DECKS MAY BE AS DUPLICATE CARD DECKS
000057                          *    OR AS MEMBER "VK" OF A MULTIPLE MEMBER FILE. IN THE MOST FREQUENT CASE
000058                          *    IT WILL BE FOUND AS MEMBER VK WITHIN FILE PROGFILE OF A DISKETTE VOLUME
000059                          *    ENTITLED DIAGS.
000060                          *
000061                          *    DISTRIBUTION OF THE LISTINGS, WHICH SHOULD BE AVAILABLE IF ANY COMPLEX
000062                          *    MAINTENANCE OR REPAIR IS TO BE PERFORMED, IS NORMALLY MADE AS A
000063                          *    PRINTED COPY.
000064                          *
000065                          *
000066                          * ROUTINE DEMONSTRATION:
000067                          *
000068                          *    A MINIMUM SATISFACTORY TEST FOR NORMAL OPERATION MAY BE OBTAINED
000069                          *    BY ENTERING THE CHANNEL NUMBER TO BE TESTED AND COMPLETING ONE PASS.
000070                          *
000071                          *
000072                          * MAIN MEMORY REQUIREMENT:
000073                          *
000074                          *    THIS PROGRAM REQUIRES 16K WORDS OF MAIN MEMORY AND WILL USE
000075                          *    ALL OF AVAILABLE MEMORY THROUGH 64K WORDS.
000076                          *
000077                          ************************************************************************
000078                          *
000079                          *
000080                          *
000081                          *
000082                          * TEST PROCEEDURE AND DESCRIPTION
000083                          *
000084                          *   THE PROGRAM AUTOMATICALLY DETERMINES WHAT CHANNELS ARE ACTIVE
000085                          *   ON THE DLCP TO BE TESTED. IT PROCEEDS TO EXCERCISE ALL MOTHER
000086                          *   BOARD FUNCTIONS THAT DO NOT REQUIRE LINE ADAPTERS. THE TEST INCLUDES
000087                          *   DLCP MEMORY, INSTRUCTION SET, AND DATA XFER TESTS. PSUEDO RANDOM
000088                          *   DATA IS TRANSFERED VIA ALL CHANNELS. THE MLCP IS MADE TO TRANSFER
000089                          *   DATA TO ALL AVAILABE MEMORY ABOVE THE PROGRAM. DLCP INTERRUPTS ARE
000090                          *   MADE ON ALL ACTIVE CHANNELS AT MULTIPLE LEVELS.
000091                          *
000092                          *   AN ACTIVE CHANNEL IS ONE WHICH EITHER HAS A LINE ADAPTER PRESENT
000093                          *   OR HAS ITS LINE ADAPTER-HERE SIGNAL TIED TO GROUND. THE  TEST
000094                          *   PROGRAM CAN ONLY TEST ACTIVE CHANNELS.
000095                          *
000096                          *
000097                          * OPERATING INSTRUCTIONS
000098                          *
000099                          *     LOAD AND START (OR RESTART) THE PROGRAM. THE PROGRAM IDENTIFICATION WILL
000100                          *     BE DISPLAYED ON THE CONSOLE.  THE INITIAL START WILL ALSO DISPLAY:
000101                          *
000102                          *       THE ZV$LIB REVISION NUMBER
000103                          *       THE ADDRESS FORM (SAF OR LAF)
000104                          *       I/O  EQUIPMENT DETECTED IN THE SYSTEM
000105                          *       MEMORY SIZE
000106                          *
000107                          *     THIS DISPLAY MUST BE VERIFIED BY THE OPERATOR.   THIS DISPLAY IS OMITTED
000108                          *     ON RESTARTS.
000109                          *
000110                          *     THE CONSOLE SEARCH RULES ARE:  FIND THE CONSOLE WITH THE LOWEST  CHANNEL
000111                          *     NUMBER CONNECTED THRU AN  MDC  CONTROLLER.  IF THERE IS NO CONSOLE ON AN
000112                          *     MDC, THEN SEARCH FOR A TERMINAL WITH THE HIGHEST CHANNEL NUMBER ASSIGNED
000113                          *     TO AN ACLA ADAPTER ON AN DLC CONTROLLER.  IF NO ASYNC ADAPTER  IS FOUND,
000114                          *     THEN GO TO THE FULL CONTROL PANEL.
000115                          *
000116                          *     THERE ARE THREE CONSOLE CHANNEL OPTIONS DETERMINED BY THE VALUE  OF  LO-
000117                          *     CATION "ZV$TTY".
000118                          *
000119                          *     IF ZV$TTY EQUALS (0000), SEARCH FOR A CONSOLE.
000120                          *     IF ZV$TTY EQUALS (FFFF), ASSUME THERE IS NO CONSOLE.
000121                          *     IF ZV$TTY EQUALS NEITHER (0000), NOR (FFFF), THEN IT IS THE CONSOLE CHAN-
000122                          *     NEL NUMBER.   NOTE:  DEFAULT IS TO SEARCH FOR A CONSOLE.
000123                          *
000124                          *     ALL CONSOLE I/O IS EVEN PARITY.  IF CONSOLE IS ON MLC,  IT MUST BE ASYNC
000125                          *     AND THE BAUD RATE SET AT 1200 TO MATCH THE PROGRAM SUPPLIED RATE.  IF IT
000126                          *     IS NECESSARY TO CHANGE THE PROGRAM BAUD RATE,  THEN THE  NEW  BAUD RATE
000127                          *     CODE SHOULD BE PUT INTO LOCATION "ZV$BUD" IN HEX. THE TERMINAL BAUD RATE
000128                          *     MUST BE  SET  TO MATCH THIS NEW BAUD RATE.  THE CORRECT HEX VALUE MAY BE
000129                          *     OBTAINED FROM THE FOLLOWING TABLE.
000130                          *
000131                          *-----------------------------------------*
000132                          *                                         *
000133                          *        BAUD  RATE  TABLE                *
000134                          *                                         *
000135                          *-----------------------------------------*
000136                          * ACLA 1.D.E(2118)(2110) E(2108)
000137                          * BAUD-RATE
000138                          *    50    E   0 EE   1
000139                          *    75    E   1  EE   2
000140                          *   110    E   2 EE   3
000141                          *   134    E   3 EE   4
000142                          *   150    E   4 EE   5
000143                          *   200    E   5 EE  ---
000144                          *   300    E   6 EE   6
000145                          *   600    E   7 EE   7
000146                          *   900    E  ---  EE   8
000147                          *  1050    E   8 EE  ---
000148                          *  1200    E   9 EE   9
```

```
000149                  *  1800    E  10 (A) EE  10 (A)
000150                  *  2000    E  11 (B) EE  ---
000151                  *  2400    E  12 (C)  E  11 (B)
000152                  *  3600    E  --- EE  12 (C)
000153                  *  4800    E  13 (D) EE  13 (D)
000154                  *  7200    E  --- EE  14 (E)
000155                  *  9600    E  14 (E) EE  15 (F)
000156                  * 19200    E  15 (F) EE  ---
000157                  *
000158                  *     TO MAKE ANY OF THE ABOVE CHANGES,  LOAD AND HALT THE PROGRAM BEFORE EX-
000159                  *     ECUTION.  INSERT CHANGE THEN EXECUTE.  MEMORY LOCATIONS OF "ZV$TTY" AND
000160                  *     "ZV$BUD" MAY BE FOUND IN MAP AT END OF LISTING.
000161                  *     CONSULT LEVEL-6 T&V MANUAL "AW94" FOR DETAILS ON HOW TO LOAD THE TESTS.
000162                  *
000163                  *     THE FOLLOWING IS A TYPICAL RESULT OF LOADING  AND  STARTING TO  RUN THE
000164                  *     PROGRAM.
000165                  *
000166                  *     DLCP TEST  DLCS1 <PGM DATE> <PGM REV>
000167                  *     ZV$LIB REV. 6.0
000168                  *     ZV$AF= 1  <2>
000169                  *     WDT
000170                  *     CHAN  DEVC  ID
000171                  *     0400  DSK1  2010
000172                  *     0480  DSK1  2010
000173                  *     0580  CDR   2008
000174                  *     1200  DISC  2330
000175                  *     1280  DISC  2330
000176                  *     1300  LPT   2000
000177                  *     1380  CONS  2019
000178                  *     MEMORY LOW  00002B2D
000179                  *     MEMORY HIGH 00003FFF  16K
000180                  *
000181                  *   THE PROGRAM WILL THEN ASK:
000182                  *
000183                  *              PWR FREQ (HZ)?:
000184                  *
000185                  *    THE OPERATOR MUST TYPE IN THE FREQUENCY OF THE CLOCK
000186                  *    USED TO DRIVE THE REAL TIME CLOCK ON HIS SYSTEM. THE
000187                  *    PROGRAM NEXT DISPLAYS:
000188                  *
000189                  *
000190                  *              DLCP CHANNEL?:
000191                  *
000192                  *    THE OPERATOR MUST RESPOND WITH THE LOWEST ACTIVE ADDRESS ON
000193                  *    THE MLCP. ENTER 4 HEX DIGITS FOLLOWED BY A CARRIAGE RETURN.
000194                  *    THE MLCP ADDRESS WILL NORMALLY BE DISPLAYED AS AN ITEM IN THE
000195                  *    CONFIGURATION PRINTOUT.
000196                  *
000197                  *    THE PROGRAM WILL THEN RUN ABOUT 1 MINUTE AND TYPE:
000198                  *
000199                  *              FW REV  XX
000200                  *
000201                  *    THIS IS A PRINTOUT OF THE CURRENT FIRMWARE REVISION.
000202                  *    THE ABOVE  PRINTOUT IS GIVEN ON THE FIRST PASS ONLY.
000203                  *    VALUE "XX" WILL BE STORED IN LOC "FW-REV" FOR EXAMINATION
000204                  *    BY USERS WITH NO CONSOLE.
000205                  *
000206                  *
000207                  *    IF THERE  IS NO CONSOLE PRESENT REFER TO THE MANUAL "SERIES 60
000208                  *    LEVEL 6 MODEL 34/36 SYSTEM COACTIVE MAINTENANCE OPERATORS
000209                  *    GUIDE" DOC 71010202-100 FOR INSTRUCTIONS ON ENTERING DATA
000210                  *    AND INTERPRETING PROGRAM MESSAGES.
000211                  *
000212                  *    THE PROGRAM WILL RUN APPROXOMATELY 3 - 5 MINUTES IF THERE ARE NO
000213                  *    HARDWARE FAULTS. IT WILL THEN DISPLAY:
000214                  *
000215                  *         PASS
000216                  *
000217                  *    AND HALT AT WITH P COUNTER DISPLAY AT HEX 100. TO  CONTINUE
000218                  *    HIT EXECUTE. THE PROGRAM WILL CONTINUE AND TYPE "PASS"
000219                  *    EVERY 3 - 5 MINUTES WITH NO HALTS BETWEEN PASSES.
000220                  *
000221                  *    TO CHANGE TO A SECOND DLCP ADDRESS RESTART THE PROGRAM AT
000222                  *    HEX 105. PROGRAM OPERATION WILL THEN PROCEED AS AFTER AN INITIAL
000223                  *    START AT HEX 100.
000224                  *
000225                  *    NOTE - FOR OPERATION WITH THE BASIC CONTROL PANEL THE PROGRAM
000226                  *           MUST BE RELOADED TO TEST A SECOND MLCP.
000227                  *
000228                  *
000229                  * ERROR REPORTS
000230                  *
000231                  *    ERRORS WILL CAUSE THE PROGRAM TO HALT. AN ERROR MESSAGE WILL
000232                  *    BE DISPLAYED IF A CONSOLE IS PRESENT. ALL ERROR MESSAGES ARE AN
000233                  *    INDICATION THAT THE COMMUNICATIONS PROCESSOR IS FAULTY.
000234                  *
000235                  *    ERROR DISPLAYS ARE AS FOLLOWS:
000236                  *
000237                  *              ERR MBXX AT YYYY
000238                  *
000239                  *              B7  B6  B5  B4  B3  B2  B1  I
000240                  *              R7  R6  R5  R4  R3  R2  R1  M
000241                  *
000242                  *         XX   = TEST LABLE. SEE JUMP TABLE AT LOCATION "BKK1"
000243                  *                FOR A LIST OF TESTS PERFORMED.
000244                  *         YYYY = ERROR LOCATION IN LISTING. HAS COMMENT
000245                  *                GIVING FAILING FUNCTION.
000246                  *
000247                  *         B1-B7, I, R1-R7, M ARE  CONTENTS OF REGISTERS.
000248                  *         INTERPERTATION OF THESE IS FOR SPECIALIST USAGE.
000249                  *
000250                  *         IN ALL CASES:
000251                  *
000252                  *              R3 = CHANNEL NUMBER
000253                  *
000254                  *         IN GENERAL
000255                  *
000256                  *              R6 = SHOULD BE DATA
000257                  *              R5 = ACTUAL DATA
000258                  *              R7 = WORD NUMBER IN BLOCK TRANSFER
000259                  *
000260                  *
000261                  *    A PROGRAM HALT WITH P DISPLAYING  A VALUE LESS THAN
```

```
000262                  *    HEX 100 INDICATES THAT A TRAP OCCURED. P DISPLAYS ONE
000263                  *    BEYOND THE ACTUAL TRAP VECTOR WHICH WAS TAKEN. THE
000264                  *    TRAP SAVE AREA IS FROM LOC. 2 - 9. REFER TO MODEL
000265                  *    34/36 T + V OPERATING INSTRUCTION MANUAL, DOC
000266                  *    71010460-200, SECTION 1.1 .
000267                  *
000268                  *    IF DURING THE OPERATION OF THE PROGRAM THE DISPLAY PANEL
000269                  *    BECOMES "HUNG" SUCH THAT THE DISPLAYED REGISTER IS
000270                  *    UNCHANGING AND NO OTHER REGISTER MAY BE SELECTED, THEN
000271                  *    THE MLCP BEING TESTED SHOULD BE REPLACED.
000272                  *
000273                  *
000274                  *
000275                  *
000276                  ****************************************
000277                  ****************************************
000278                  *
000279                  * RESTRICTIONS:
000280                  *
000281                  *    FOR SYSTEMS WITH FIRMWARE REV NUMBER 4 OR LESS, REV. A
000282                  *    OF THE PROGRAM MUST BE USED.
000283                  *
000284                  *    THE FIRMWARE REV. NUMBER AT THE TIME OF RELEASING
000285                  *    REV. B OF THIS PROGRAM IS "5".
000286                  *
000287                  *    THE FIRMWARE REV NUMBER AT THE TIME OF RELEASING REV C
000288                  *    OF THE PROGRAM IS "8".
000289                  *
000290                  *
000291                  *    THE FIRMWARE REV. NUMBER AT THE TIME OF RELEASING REV D
000292                  *    OF THE PROGRAM IS "9". REV D CONTAINS NO FUNCTIONAL
000293                  *    DIFFERENCES FOR SAF MODE.
000294                  *
000295                  *
000296                  *    IF THIS PROGRAM IS TO BE RUN ON A SYSTEM WITH A BASIC CONTROL
000297                  *    PANEL, A SYSTEM CONSOLE MUST BE PRESENT.
000298                  *
000299                  ****************************************
000300                  ****************************************
000301                  *
000302                  *
000303                  ************************************************************************
```

```
000304                                          /
000305                                          *
000306                                          *
000307                                          *       STANDARD REGISTER ASSIGNMENTS
000308                                          *
000309                                          *             $B5          V$LIB ENTRY
000310                                          *             $B2          ENTRY TO MAJOR TEST
000311                                          *             $B4,$B1      ENTRY TO SUB-ROUTINES
000312                                          *E    $B7E       ENTRY TO ERROR ROUTINE
000313                                          *
000314                                          *E    $R3          CHANNEL
000315                                          *E    $R5E       'IS' DATA WORD
000316                                          *E    $R6E       'SHOULD BE' DATA WORD
000317                                          *E    $R7E       TABLE COUNT
000318                                          *
000319                                          *
000320          0000                            ZERO   EQU    $
000321                                                 XLOC   ZHNTSA,ZHTH15,ZV$HR
000322                                                 XLOC   ZV$RD,ZHTSA,ZHTH17
000323                                                 XLOC   ZHISAZ,ZHIAFB
000324                                                 XLOC   ZHWDTC,ZHRTCC
000325                                                 XLOC   ZHRTC1,ZHRTCL
000326                                                 XLOC   ZHPFR,ZHCOMM
000327                                                 CTRL   LINK ZV$TH
000328                                          *
000329                                          *
000330                                          *
000331                                          *
000332                                          *
000333                                          *
000334  00FF                                           ORG    ZERO+X'FF'
000335  00FF  0000                              STOP   HLT                     NO ERROR FOUND IN MLCC
000336                                          *
000337                                          *             OPERATOR MUST ENTER MLCC CHANNEL
000338                                          *
000339  0100  89C0 13A9                        STRT   CMZ    FRST             CHECK FIRST  TIME FLAG
000340  0102  0981 00BA                               BNE    STLOOP           BRANCH IF NOT FIRST TIME
000341  0104  0F7C                                    NOP    >STRT
000342                                          *
000343  0105  8700 14AC                        RESTRT CL     <PASS
000344  0107  8700 00FF                               CL     <STOP            INSERT FIRST TIME THRU HALT
000345  0109  0F82                              DRSR1  B      >DRSR            BECOMES NOP AFTER FIRST TIME
000346  010A  0F8B                                     B      >RST             BYPASS ZV$RD
000347                                          *
000348                                          DRSR   CALL   ZV$RD,MESG
        010B  FBC0 0003
        010D  D380 0000      X
        010F  0F80
        0110  2701
000349  0111  9870 0F02                               LDR    $R1,=Z'0F02'     NOP
000350  0113  9F00 0109                               STR    $R1,<DRSR1
000351  0115  8980 14AA                        RST    CMZ    <FRST
000352  0117  0980 0128                               BNE    <RST1            BRANCH IF NOT FIRST1 TIME
000353  0119  1C3C                                    LDV    $R1,=60
000354  011A  9F00 154E                               STR    $R1,<HRTZ        DEFAULT FREQ IS 60
000355                                                 CALL   ZV$WC,MESG5
        011C  FBC0 0003
        011E  D380 0000      X
        0120  0F80
        0121  2718
000356                                                 CALL   ZV$ID,HRTZ       INPUT FREQUENCY
        0122  FBC0 0003
        0124  D380 0000      X
        0126  0F80
        0127  154E
000357                                          RST1   CALL   ZV$WC,MESG1
        0128  FBC0 0003
        012A  D380 0000      X
        012C  0F80
        012D  1567
000358                                                 CALL   ZV$IH,DUMMY
        012E  FBC0 0003
        0130  D380 0000      X
        0132  0F80
        0133  14C7
000359  0134  9800 14C7                               LDR    $R1,<DUMMY
000360  0136  9570 FFC0                               AND    $R1,=Z'FFC0'     STRIP TO CHANNEL NUMBER ONLY
000361  0138  CB80 0000      X                        LAB    $B4,<ZHTSA       SET UP FOR TRAP ON
000362  013A  CF80 0000      X                        STB    $B4,<ZHNTSA      UNAVAILIABLE RESOURCE
000363  013C  CB80 0156                               LAB    $B4,<NODEVF
000364  013E  CF80 0000      X                        STB    $B4,<ZHTH15
000365  0140  9970 03C0                               CMR    $R1,=X'3C0'      COMPARE CHANNEL WITH HEX 3C0
000366  0142  0302                                    BG     >CON1            BRANCH IF GREATER
000367  0143  0F8A                                    B      >JP1             CHANNEL NUMBER TO SMALL
000368  0144  1E26                              CON1   ADV    $R1,=X'26'       PUT FUNCTION IN
000369  0145  8055                                    IO     =$R5,=$R1        INPUT DEVICE NUMBER
        0146  0051
000370  0147  C855                                    LDR    $R4,=$R5
000371  0148  D570 FF00                               AND    $R5,=Z'FF00'
000372  014A  D970 3100                               CMR    $R5,=Z'3100'     COMPARE DEVICE NUMBER RESPONSE
000373  014C  090C                                    BE     >CON2
000374                                          *
000375                                          JP1    CALL   ZV$ER,$,ERMG1    MLCC NOT FOUND ON THIS ADDRESS
        014D  FBC0 0003
        014F  D380 0000      X
        0151  0F80
        0152  014D
        0153  1583
000376  0154  0F80 0115                               B      <RST
000377                                          *
000378  0156  8755                              NODEVF CL     =$R5             TRAP HANDLER ROUTINE
000379  0157  0003                                    RTT
000380                                          *
000381                                          *
000382                                          *
000383                                          *       PREPARE ACTIVE LINE TABLE
000384                                          *
000385                                          *
000386                                          CON2   CALL   ZV$F,ATLT,ALLONE,C16
        0158  FBC0 0003
        015A  D380 0000      X
        015C  0F80
        015D  14AD
```

```
                 015E  14A8
                 015F  14A9
000387           0160  8700 14C9              CL      <CH-CT          CHANNEL COUNT
000388           0162  8700 14BD              CL      <IMASK          SET LINE MASK WORD TO 0
000389                                  *
000390           0164  0854                     LDR     $R5,=$R4
000391           0165  B851                     LDR     $R3,=$R1
000392           0166  B570 03C0               AND     $R3,=X'3C0'     STRIP $R3 TO SUBCH. NUMBER
000393           0168  3046                     SOR     $R3,6
000394           0169  9570 FC3F               AND     $R1,=Z'FC3F'    STRIP IO CONTROL WORD WITHOUT SUBCH.
000395           016B  DF30 14AD        CON2A   STR     $R5,<ATLT.$R3   STORE ID IF SUBCH. PRESENT
000396           016D  BF00 14CD                STR     $R3,<TPR
000397           016F  D970 3178                CMR     $R5,=X'3178'
000398           0171  09B0                     BNE     >CON2C
000399                                  *
000400                                  * READ EXTENDED ID
000401                                  *
000402           0172  C851                     LDR     $R4,=$R1
000403           0173  4E02                     ADV     $R4,=2          FORM FC FOR EXT ID
000404           0174  C380 1173                LNJ     $B4,<CGSCH
000405           0176  8055                     IO      =$R5,=$R4
                 0177  0054
000406           0178  0703                     BIOT    >$+2+$AF
000407           0179  F380 125C                LNJ     $B7,<ERROR      INPUT EXTEND ID WAS NAK'ED
000408           017B  DF30 14AD                STR     $R5,<ATLT.$R3
000409           017D  3B80 01A1                BUDD    $R3,<CON2C
000410                                          CALL    ZV$IC,IDMSG     LINE
                 017F  FBC0 0003
                 0181  D380 0000        X
                 0183  0F80
                 0184  1570
000411           0185  3041                     SOR     $R3,1           GET LINE NUMBER
000412           0186  BF00 2BAA                STR     $R3,<RTB
000413                                          CALL    ZV$TD,RTB       NUMBER
                 0188  FBC0 0003
                 018A  D380 0000        X
                 018C  0F80
                 018D  2BAA
000414           018E  B800 14CD                LDR     $R3,<TPR        GET BACK CHANNEL
000415                                          CALL    ZV$T,EQ         EQUALS
                 0190  FBC0 0003
                 0192  D380 0000        X
                 0194  0F80
                 0195  1574
000416           0196  5048                     SOR     $R5,8           ALIGN EXTENDED ID
000417           0197  D470 3100                OR      $R5,=Z'3100'    PUT IN MLCP ADDRESS
000418           0199  DF00 14CD                STR     $R5,<TPR
000419                                          CALL    ZV$IH,TPR       PRINT ID
                 019B  FBC0 0003
                 019D  D380 0000        X
                 019F  0F80
                 01A0  14CD
000420           01A1  8930 14BD        CON2C   LDT     <IMASK.$R3      SET MASK WORD OF ACTIVE LINES
000421           01A3  8AD3             CON2B   INC     =$R3
000422           01A4  3D10                     CMV     $R3,=16         SEE IF ALL SUBCH. TESTED
000423           01A5  0909                     BE      >CON3
000424           01A6  C851                     LDR     $R4,=$R1
000425           01A7  C380 1173                LNJ     $B4,<CGSCH
000426           01A9  8055                     IO      =$R5,=$R4       INPUT ID
                 01AA  0054
000427           01AB  0701 FFBF                BIOT    CON2A
000428           01AD  0FF6                     B       >CON2B
000429                                  *
000430                                  *
000431           01AE  2CF5             CON3    LDV     $R2,=-11        NUMBER OF I/O TO BE CHANGED
000432           01AF  DB80 1567                LAB     $B5,<CONT1+11   PUT ADDRESS OF CONTROL TABLE IN $B5
000433           01B1  C870 003F        ALL     LDR     $R4,=X'3F'      LOAD MASK TO CLEAR CHANNEL
000434           01B3  9570 FC00                AND     $R1,=Z'FC00'    CLEAR SUBCH. & FUNCTION
000435           01B5  9F00 14C3                STR     $R1,<MLCADD     STORE MLC ADDRESS
000436           01B7  C525                     AND     $R4,$B5.$R2     CLEAR CHANNEL
000437           01B8  9454                     OR      $R1,=$R4        PUT CHANNEL NUMBER IN
000438           01B9  9F25                     STR     $R1,$B5.$R2     STORE IT BACK IN CONTROL TABLE
000439           01BA  27F7                     BINC    $R2,>ALL
000440           01BB  0F80 01CA                B       <BKK1
000441                                  *
000442                                  *
000443                                  *
000444                                  *
000445                                  *
000446           01BD  9800 01C9        STLOOP  LDR     $R1,<NOP
000447           01BF  9F00 00FF                STR     $R1,<STOP       NO HALTS AFTER FIRST PASS
000448           01C1  8700 14AB                CL      <LOOP           SET LOOP COUNT TO 0
000449           01C3  8700 1549                CL      <SEC            CLEAR SECONDS COUNTER
000450           01C5  8700 154A                CL      <TTOT           CLEAR TIME TOTALIZER
000451           01C7  0F80 01CA                B       <BKK1
000452                                  *
000453                                  * JUMP TABLE FOLLOWS. THIS IS THE CONTROL SECTION OF THE PROGRAM.
000454                                  *
000455                                  *                               LABLE  -  DESCRIPTION
000456           01C9  0F03             NOP     NOP     >BKK3
000457           01CA  A380 021E        BKK1    LNJ     $B2,<RGNT        A - GO TO RANGE TEST
000458           01CC  A380 02D6        BKK3    LNJ     $B2,<CCBRT       B - GO TO CCB READ TEST
000459           01CE  A380 0355        BKK4    LNJ     $B2,<MEMT        C - GO TO MEMORY TEST
000460           01D0  A380 0470        BKK4B   LNJ     $B2,<LCTOB       E - GO TO OUTPUT BYTE AND LCT TEST
000461           01D2  A380 0539                LNJ     $B2,<SITEST      SI - SOFT INITIALIZE TEST
000462           01D4  A380 0639                LNJ     $B2,<INSTT       GO TO INSTRUCTION TEST
000463           01D6  A380 06B5        BKK7    LNJ     $B2,<INCONT      H - OUTPUT INTERRUPT CONTROL TEST
000464           01D8  A380 0815        BKK7B   LNJ     $B2,<STIORC      K - STOP I/O TEST
000465           01DA  A380 0C39                LNJ     $B2,<INST1       R - FIRST INTERRUPT TEST
000466           01DC  A380 0CEF        BKK9B   LNJ     $B2,<DSST        S - DATA SET SCAN TEST
000467           01DE  A380 0E3F        BKK10   LNJ     $B2,<SITST       T - DEFERRED INTERRUPT TEST
000468           01E0  A380 086E        BKK7C   LNJ     $B2,<RCO         L - RECIEVE TEST
000469           01E2  A380 08BC        BKK7D   LNJ     $B2,<TRAN1       M - TRANSMIT TEST
000470           01E4  A380 0436        BKK4A   LNJ     $B2,<MNKT        C - GO TO MEMORY NAK TEST
000471           01E6  A380 05B7        BKK5    LNJ     $B2,<CHPGT       F- GO TO CHANNEL PROGRAM TEST
000472           01E8  A380 0AD6                LNJ     $B2,<CRCT        GO TO CRC TEST
000473           01EA  A380 0BBC        BKK9    LNJ     $B2,<OTPT        GO TO OUTPUT TEST
000474           01EC  A380 06ED        BKK7A   LNJ     $B2,<IZTST       CHANNEL INITIALIZE TEST
000475                                          CALL    ZV$T,ZV$TC,DON
                 01EE  FBC0 0003
                 01F0  D380 0000        X
                 01F2  0F80
                 01F3  154F
```

```
000476  01F4  0F81 FF0A              B       STOP
000477                               CALL    ZV$T,ZV$TC,DONE
        01F6  FBC0 0003
        01F8  D380 0000      X
        01FA  0F80
        01FB  1578
000478  01FC  0F81 FF02              B       STOP
000479  01FE  0000                   HLT
000480                          *
000481                          * ACCUMULATE TIME
000482  01FF  9800 1549              LDR     $R1,<SEC                GET NUMBER OF SECONDS
000483  0201  9970 00B4              CMR     $R1,=180                CHECK IF 3 MIN
000484  0203  0287                   BGE     >PSPT                   BRANCH MEANS > OR = 3 MIN
000485  0204  0F80 01CC              B       <BKK3
000486                               CALL    ZV$PCH
        0206  FBF0 0001
        0208  D380 0000      X
000487                          PSPT CALL    ZV$TC,PASMSG
        020A  FBC0 0003
        020C  D380 0000      X
        020E  0F80
        020F  021B
000488  0210  8A80 14AC              INC     <PASS                   INCREMENT PASS COUNT
000489  0212  9800 14AC              LDR     $R1,<PASS
000490  0214  1D01                   CMV     $R1,=1
000491  0215  0980 0100              BNE     <STRT                   NOT FIRST PASS
000492  0217  8700 00FF              CL      <STOP                   FIRST PASS
000493  0219  0F80 00FF              B       <STOP
000494  021B  5041 5353 2024  PASMSG TEXT    'PASS $'
000495                          *
000496                          *-----------------------------------------------------------
000497                          *
000498                          *       RANGE TEST
000499  021E  9870 4120          RGNT LDR    $R1,=A'A '
000500  0220  9F00 1581              STR     $R1,<ERMG+1
000501                               CALL    ZV$T,ZV$TC,TSTA
        0222  FBC0 0003
        0224  D380 0000      X
        0226  0F80
        0227  1590
000502  0228  9380 0F18              LNJ     $B1,<INTITZ             INITIALIZE TRAP + RUPT ERROR VECTORS
000503  022A  6C01                   LDV     $R6,=1                  INITIALIZE RANGE
000504  022B  C380 10FC              LNJ     $B4,<GENITZ             GENERAL INITIALIZE TO MLCC
000505                          *
000506  022D  9870 0300              LDR     $R1,=X'300'             LOAD IMGA WITH HEX 300
000507  022F  9F00 14DA              STR     $R1,<IMGA+($AF-1)
000508  0231  8751                   CL      =$R1                    CLEAR $R1
000509  0232  8753                   CL      =$R3                    FIRST CHANNEL
000510  0233  C380 10EC          NXCH LNJ    $B4,<FLN                FIND LINE ON
000511  0235  0FA3                   B       >JMPEXT                 GO DO SECOND HALF OF TEST
000512                          *
000513  0236  4C0A                   LDV     $R4,=10
000514  0237  CA00 1549              ADD     $R4,<SEC                UPDATE TIME, 10 SEC/CHAN
000515  0239  CF00 1549              STR     $R4,<SEC
000516  023B  C800 155C          NXRG LDR    $R4,<CONT1              LOAD $R4 WITH I/O CONTROL WORD
000517  023D  C380 1173              LNJ     $B4,<CGSCH              PUT CHANNEL IN I/O CONTROL WORD
000518  023F  81C8 129A              IOLD    *IMGA,=$R4,=$R6         OUTPUT ADDRESS AND RANGE
        0241  0054
        0242  0056
000519  0243  0703                   BIOT    >IOT1
000520  0244  F380 125C              LNJ     $B7,<ERROR              I/O TRANSFER DID NOT TAKE PLACE
000521                          *
000522  0246  4E06              IOT1 ADV     $R4,=6                  OUTPUT CCB CONTROL FIELD
000523  0247  8070 0000              IO      =0,=$R4
        0249  0054
000524  024A  0703                   BIOT    >$+2+$AF
000525  024B  F380 125C              LNJ     $B7,<ERROR
000526                          *
000527  024D  C800 155F              LDR     $R4,<CONT4              INPUT NXT STAT TO ADVANCE POINTER
000528  024F  C380 1173              LNJ     $B4,<CGSCH
000529  0251  8040 1275              IO      DUMMY,=$R4              INPUT NXT STAT
        0253  0054
000530  0254  0F85                   B       >IOT3
000531  0255  F380 125C              LNJ     $B7,<ERROR              INSTRUCTION WAS NACKED
000532  0257  0F82                   B       >IOT3
000533  0258  0F96              JMPEXT B     >SCCBTS
000534                          *
000535  0259  C800 155D          IOT3 LDR    $R4,<CONT2              LOAD $R4 WITH I/O CONTROL WORD
000536  025B  C380 1173              LNJ     $B4,<CGSCH              I/O CONTROL WORD IN $R4
000537  025D  8055                   IO      =$R5,=$R4               INPUT RANGE
        025E  0054
000538  025F  0703                   BIOT    >IOT2
000539  0260  F380 125C              LNJ     $B7,<ERROR              I/O TRANSFER DID NOT TAKE PLACE
000540  0262  E955               IOT2 CMR    $R6,=$R5                COMPARE OUTPUT RANGE WITH INPUT RANGE
000541  0263  0988                   BNE     >RGE                    NOT EQUAL BRANCH TO RGE
000542                          *
000543                          *
000544  0264  EA70 000A          RGNT1 ADD    $R6,=Z'000A'            SET FOR EXT RANGE
000545  0266  0680 023B              BCF     <NXRG
000546  0268  8AD3                   INC     =$R3                    ADD ONE TO CHANNEL
000547  0269  0F80 0233              B       <NXCH                   BRANCH TO NXCH
000548                          *
000549  026B  F380 125C          RGE  LNJ    $B7,<ERROR              RANGE ERROR
000550  026D  0FF7                   B       >RGNT1                  CONTINUE TESTING
000551                          *
000552                          * TEST CCB'S DON'T OVERWRITE EACH OTHER
000553                          *
000554  026E  C380 10FC          SCCBTS LNJ  $B4,<GENITZ             GENERAL INITIALIZE
000555  0270  2C01                   LDV     $R2,=1                  INITIAL DATA
000556  0271  F870 0200              LDR     $R7,=X'200'             DATA FOR CONTROL FLAG
000557  0273  8753                   CL      =$R3
000558  0274  C380 10EC          NLINE LNJ   $B4,<FLN
000559  0276  0FA2                   B       >RCCBS                  NO MORE LINES
000560  0277  1CFC                   LDV     $R1,=-4                 COUNTS 4 CCB'S
000561  0278  C800 155C          NXTCCB LDR  $R4,<CONT1              IOLD FUNCTION CODE
000562  027A  C380 1173              LNJ     $B4,<CGSCH              FORM I/O CONTROL WORD
000563  027C  81C8 125D              IOLD    *IMGA,=$R4,=$R2
        027E  0054
        027F  0052
000564  0280  0703                   BIOT    >$+2+$AF
000565  0281  F380 125C              LNJ     $B7,<ERROR              IOLD WAS NACKED
000566  0283  4E06                   ADV     $R4,=6                  FORM CCB CONTROL WORD
000567  0284  8057                   IO      =$R7,=$R4               OUTPUT CONTROL FLAG
```

```
                0285  0054
000568          0286  0703                    BIOT    >$+2+$AF
000569          0287  F380 125C                LNJ     $B7,<ERROR
000570          0289  8AD2                    INC     =$R2                    BUMP FOR NEXT RANGE
000571          028A  FA70 0100                ADD     $R7,=X'100'             FORM NEXT CONTROL WORD
000572          028C  17EC                    BINC    $R1,>NXTCCB             BRANCH MEANS THIS CHAN NOT DONE
000573          028D  C270 0006                SUB     $R4,=Z'0006'
000574          028F  81C8 124A                IOLD    *IMGA,=$R4,=$R2         GIVE  A 5TH IOLD
                0291  0054
                0292  0052
000575          0293  0783                    BIOF    >$+2+$AF                IT SHOULD BE NAK'ED
000576          0294  F380 125C-               LNJ     $B7,<ERROR              5TH IOLD WASN'T NAK'ED
000577          0296  8AD3                    INC     =$R3                    BUMP CHANNEL LINE
000578          0297  0FDD                    B       >NLINE
000579                                *
000580                                * READ RANGES SET UP
000581                                *
000582          0298  2C01            RCCBS   LDV     $R2,=1                  INITIAL DATA
000583          0299  F870 0200                LDR     $R7,=X'200'             DATA FOR STATUS
000584          029B  8753                    CL      =$R3
000585          029C  C380 10EC        NL1     LNJ     $B4,<FLN                FIND LINE NUMBER
000586          029E  8382                    JMP     $B2                     ALL DONE, EXIT TEST
000587          029F  1CFC                    LDV     $R1,=-4                 COUNTER FOR 4 CCB'S
000588          02A0  C800 155F        RNXR    LDR     $R4,<CONT4
000589          02A2  C380 1173                LNJ     $B4,<CGSCH              FORM IO CONTROL WORD
000590          02A4  8040 1222                IO      DUMMY,=$R4              INPUT NEXT STATUS
                02A6  0054
000591          02A7  0703                    BIOT    >$+2+$AF
000592          02A8  F380 125C                LNJ     $B7,<ERROR              INPUT NXT STATUS WAS NAK'ED
000593          02AA  C800 155D                LDR     $R4,<CONT2
000594          02AC  C380 1173                LNJ     $B4,<CGSCH              FORM IO CONTROL WORD
000595          02AE  8055                    IO      =$R5,=$R4               INPUT RANGE
                02AF  0054
000596          02B0  0703                    BIOT    >$+2+$AF
000597          02B1  F380 125C                LNJ     $B7,<ERROR
000598          02B3  E852                    LDR     $R6,=$R2                GET SHOULD-BE DATA TO R6
000599          02B4  D956                    CMR     $R5,=$R6
000600          02B5  0903                    BE      >$+2+$AF
000601          02B6  F380 125C                LNJ     $B7,<ERROR              RANGE READ IN IS WRONG
000602                                * DO AN INPUT STATUS. IF IT ADVANCES THE INPUT POINTER A FAILURE WILL RESULT.
000603          02B8  C800 1561                LDR     $R4,<CONT6              INPUT STATUS CFUNCTION
000604          02BA  C380 1173                LNJ     $B4,<CGSCH              FORM I/O CONTROL WORD
000605          02BC  8055                    IO      =$R5,=$R4               INPUT STATUS
                02BD  0054
000606          02BE  0703                    BIOT    >$+2+$AF
000607          02BF  F380 125C                LNJ     $B7,<ERROR              INPUT STATUS WAS NAK'ED
000608          02C1  E857                    LDR     $R6,=$R7                GET SHOULD BE DATA TO R6
000609          02C2  D956                    CMR     $R5,=$R6
000610          02C3  0903                    BE      >$+2+$AF
000611          02C4  F380 125C                LNJ     $B7,<ERROR              STATUS READ IN INCORRECTLY
000612                                *
000613          02C6  8AD2                    INC     =$R2                    BUMP RANGE
000614          02C7  FA70 0100                ADD     $R7,=X'0100'            FORM NEXT STATUS
000615          02C9  17D7                    BINC    $R1,>RNXR               READ NEXT RANGE
000616                                *
000617                                *EGIVE A 5TH INPUT NEXT STATUS AND CHECK ITS NAK'ED
000618                                *
000619          02CA  C800 155F                LDR     $R4,<CONT4              FORM I/O CONTROL
000620          02CC  C380 1173                LNJ     $B4,<CGSCH              WORD FOR INPUT NEXT STATUS
000621          02CE  8055                    IO      =$R5,=$R4               SHOULD BE NAK'ED
                02CF  0054
000622          02D0  0783                    BIOF    >$+2+$AF
000623                                *
000624          02D1  F380 125C                LNJ     $B7,<ERROR              INPUT NEXT STATUS NOT NAK'ED
000625                                *
000626          02D3  8AD3                    INC     =$R3
000627          02D4  0F81 FFC7                B       NL1                     FIND NEXT LINE NUMBER
000628                                *-------------------------------------------------------------
000629                                *
000630                                *
000631                                *
000632                                *           CCB READ TEST - TESTS OUTPUT OF CCB & BLOCK READ
000633                                *
000634                                *
000635          02D6  9870 4220        CCBRT   LDR     $R1,=A'B '
000636          02D8  9F00 1581                STR     $R1,<ERMG+1
000637                                         CALL    ZV$T,ZV$TC,TSTB
                02DA  FBC0 0003
                02DC  D380 0000        X
                02DE  0F80
                02DF  1597
000638          02E0  8753                    CL      =$R3
000639          02E1  C380 10FC        NXCB    LNJ     $B4,<GENITZ             GENERAL INITIALIZE
000640                                *
000641          02E3  9800 14DC                LDR     $R1,<SBCK+$AF-1         GET ADDRESS CONSTANT LSB
000642          02E5  9F00 14DD                STR     $R1,<SBCCB
000643          02E7  8751            NXCB1   CL      =$R1                    CLEAR INDEX FOR OUTPUT ADDRESS
000644          02E8  8700 14E1                CL      <MASK                   LOAD ZERO INTO MASK
000645          02EA  4C08                    LDV     $R4,=8                  LOAD $R4 WITH BYTE RANGE OF CCB
000646          02EB  CF00 14D5                STR     $R4,<RANGE
000647          02ED  5C04                    LDV     $R5,=4                  LOAD RANGEW WITH WORD RANGE OF CCB
000648          02EE  DF00 14D7                STR     $R5,<RANGEW
000649          02F0  C380 10EC                LNJ     $B4,<FLN                FIND LINE ON
000650          02F2  8382                    JMP     $B2                     NO LINE ON JUMP OUT OF TEST
000651          02F3  BF00 2BAA                STR     $R3,<RTB                RE-INITIALIZE RTB
000652          02F5  BF00 2BAB                STR     $R3,<RTB+1
000653          02F7  BF00 2BAC                STR     $R3,<RTB+2
000654          02F9  BF00 2BAD                STR     $R3,<RTB+3
000655                                *
000656          02FB  D853                    LDR     $R5,=$R3                FIND ADDRESS FOR BLOCK READ
000657          02FC  5F20                    MLV     $R5,=X'20'              R5 = CHAN X 20
000658          02FD  DA70 1210                ADD     $R5,=X'1210'            LOCATION OF CCB TO READ
000659          02FF  BF00 14CD                STR     $R3,<TPR                STORE CHAN NUMB
000660          0301  3B03                    BEVN    $R3,>CBRT3              BRANCH IF READ
000661          0302  88D3                    DEC     =$R3                    FORM READ CHANNEL
000662          0303  5EF8                    ADV     $R5,=-8                 FORM CCB ADDRESS FOR WRITE CHANNEL
000663          0304  BF00 0354        CBRT3   STR     $R3,<READ               STORE READ CHANNEL
000664          0306  DF00 030D                STR     $R5,<CBRT1
000665          0308  C800 14D5                LDR     $R4,<RANGE
000666                                *
000667          030A  C380 1189                LNJ     $B4,<MCCB               FORM CCB FOR BLOCK READ
000668          030C  2BAA                    DC      <RTB
000669          030D  0000            CBRT1   DC      X'0'                    RAM ADDRESS TO READ
```

```
000670                                  *
000671                                  *
000672  030E  B800 14CD                 LDR   $R3,<TPR            GET CHANNEL UNDER TEST
000673  0310  C800 14D5                 LDR   $R4,<RANGE          FORM CCB TO READ BACK
000674  0312  C380 1189          NXCB2  LNJ   $B4,<MCCB
000675  0314  1547                     DC    <RMINST
000676  0315  55AA                     DC    X'55AA'
000677                                  *
000678  0316  B800 0354          NXCB3  LDR   $R3,<READ           GET READ CHANNEL NUMBER
000679  0318  C800 1562                 LDR   $R4,<CONT7          LOAD $R4 WITH I/O CONTROL WORD
000680  031A  C380 1173                 LNJ   $B4,<CGSCH          PUT I/O CONTROL WORD IN $R4
000681  031C  8070 0800                 IO    =X'800',=$R4        READ BLOCK IN FROM MLCC
        031E  0054
000682  031F  0703                      BIOT  >$+2+$AF
000683  0320  F380 125C                 LNJ   $B7,<ERROR
000684                                  *
000685                                  * READ BACK STATUS AFTER BLOCK READ TO MAKE SURE IT'S CORRECT.
000686                                  *
000687  0322  C380 1205                 LNJ   $B4,<DLAY           TIME DELAY
000688                                  *
000689  0324  C380 1161                 LNJ   $B4,<INXT           INPUT NEXT STATUS
000690  0326  E870 1000                 LDR   $R6,=X'1000'        SHOULD BE STATUS. CCB COMPLETE ON.
000691  0328  D956                      CMR   $R5,=$R6
000692  0329  0903                      BE    >$+2+$AF
000693  032A  F380 125C                 LNJ   $B7,<ERROR          STATUS WRONG AFTER BLOCK READ
000694  032C  B800 14CD                 LDR   $R3,<TPR            GET BACK CHANNEL UNDER TEST
000695                                  *
000696                                  CALL  ZV$C,SBCCB,RTB,MASK,RANGEW,ERAR
        032E  FBC0 0003
        0330  D380 0000           X
        0332  0F80
        0333  14DD
        0334  2BAA
        0335  14E1
        0336  14D7
        0337  14E3
000697  0338  89C0 11AA                 CMZ   ERAR                IF ERROR HALT
000698  033A  090F                      BE    >NOER1
000699  033B  C800 030D                 LDR   $R4,<CBRT1          STARTING LOCATION OF CCB IN RAM
000700  033D  D800 14E5                 LDR   $R5,<ERAR+2
000701  033F  E800 14E4                 LDR   $R6,<ERAR+1
000702  0341  F800 14E3                 LDR   $R7,<ERAR
000703  0343  F380 125C                 LNJ   $B7,<ERROR          BLOCK READ ERROR
000704                                  CALL  ZV$CO               CONTINUE LOOKING FOR BLOCK READ ERROR
        0345  FBF0 0001
        0347  D380 0000           X
000705  0349  8980 14AA          NOER1  CMZ   <FRST               CHECK IF FIRST TIME THRU
000706  034B  0981 0003                 BNE   BLUG
000707  034D  C380 26C0                 LNJ   $B4,<PREV           PRINT FIRMWARE REVISION
000708  034F  8A80 14AA          BLUG   INC   <FRST               SET FIRST TIME FLAG
000709  0351  8AD3                      INC   =$R3                INCREMENT CHANNEL NUMBER
000710                                  *
000711                                  *
000712  0352  0F80 02E1                 B     <NXCB
000713  0354  0000               READ   DC    0                  READ CHANNEL STORED HERE
000714                                  *----------------------------------------------------------
000715                                  *
000716                                  *
000717                                  *
000718                                  *
000719                                  *          MLCC MEMORY TEST
000720                                  *
000721                                  *
000722  0355  9870 4320          MEMT   LDR   $R1,=A'C  '
000723  0357  9F00 1581                 STR   $R1,<ERMG+1
000724                                  CALL  ZV$T,ZV$TC,TSTC
        0359  FBC0 0003
        035B  D380 0000           X
        035D  0F80
        035E  15A2
000725  035F  8700 14E1                 CL    <MASK               INITIALIZE MASK FOR COMPARISON
000726  0361  C380 10FC                 LNJ   $B4,<GENITZ         GENERAL INITIALIZE
000727                                  *
000728  0363  C870 5555                 LDR   $R4,=X'5555'        INITIALIZE MASK1 FOR RETURN BLOCK
000729  0365  CF00 14E2                 STR   $R4,<MASK1
000730  0367  8753                      CL    =$R3
000731  0368  C380 10EC          MEMTA  LNJ   $B4,<FLN            FIND LINE ON
000732  036A  8382                      JMP   $B2
000733                                  *
000734  036B  3B00 036F                 BEVN  $R3,<MEMTB          TEST ON A LINE, NOT CHANNEL BASIS
000735  036D  8AD3                      INC   =$R3
000736  036E  0FFA                      B     >MEMTA              FIND NEXT LINE
000737  036F  8751               MEMTB  CL    =$R1                CLEAR INDEX FOR OUTPUT ADDRESS
000738  0370  8754                      CL    =$R4
000739  0371  E380 0380                 LNJ   $B6,<TESM           TEST MEMORY WITH ALL BITS NOT SET
000740  0373  4CFF                      LDV   $R4,=-1
000741  0374  E380 0380                 LNJ   $B6,<TESM           TEST MEMORY WITH ALL BITS SET
000742  0376  C870 0110                 LDR   $R4,=X'110'         TEST MEMORY BY FLOATING A
000743  0378  E380 0380                 LNJ   $B6,<TESM           ONE THROUGH A FIELD OF ZEROS
000744  037A  C870 FEEF                 LDR   $R4,=Z'FEEF'        TEST MEMORY BY FLOATING A
000745  037C  E380 0380                 LNJ   $B6,<TESM           ZERO THROUGH A FIELD OF ONE'S
000746  037E  8AD3                      INC   =$R3                SET FOR NEXT CHANNEL
000747  037F  0FE9                      B     >MEMTA              TRY NEXT LINE
000748                                  *
000749                                  *
000750  0380  2C01               TESM   LDV   $R2,=1              INITIALIZE $R2
000751  0381  E870 0300                 LDR   $R6,=X'300'         LOAD NUMBER OF WORDS INTO RANGEW
000752  0383  EF00 14D7                 STR   $R6,<RANGEW
000753  0385  8700 14D8                 CL    <RAMAD              INITIALIZE RAM ADDRESS .
000754                                  *
000755  0387  9380 0411                 LNJ   $B1,<SET            FILL READ AREA WITH SET PATTERN
000756                                  *
000757  0389  BB80 1EBB                 LAB   $B3,<SDB            LOAD $B3 WITH ADDRESS OF SDB
000758  038B  E800 14D7                 LDR   $R6,<RANGEW         INITIALIZE LOOP COUNTER
000759  038D  8256                      NEG   =$R6
000760  038E  CF73               FILL   STR   $R4,+$B3            STORE PATTERN INTO SDB
000761  038F  4011                      SCL   $R4,1               ROTATE TO FORM FULL PATTERN
000762  0390  67FE                      BINC  $R6,>FILL
000763                                  *
000764  0391  BF00 14CF                 STR   $R3,<TEMP1
000765  0393  BF00 14CD                 STR   $R3,<IPR
000766  0395  3B85                      BODD  $R3,>MTM6           FIND WRITE SUBCH.
000767  0396  8A80 14CD                 INC   <TPR
```

```
000768  0398  BE00 14CD          SWR   $R3,<TPR
000769  039A  9380 1069  MTM6    LNJ   $B1,<SDATA          BLOCK WRITE DATA PATTERN TO RAM
000770  039C  1EBB              DC    <SDB
000771  039D  0600              DC    X'600'
000772  039E  0200              DC    X'200'              RAM ADDRESS
000773  039F  8000              DC    Z'8000'             EVEN BYTE ADDRESS, 250 MS DELAY
000774                *
000775                *
000776  03A0  BE00 14CD          SWR   $R3,<TPR            GET READ SUBCH.
000777  03A2  9380 1098          LNJ   $B1,<RDATA          READ DATA PATTERN FROM RAM
000778  03A4  2BAA              DC    <RTB
000779  03A5  0600              DC    X'600'
000780  03A6  0200              DC    X'200'              RAM ADDRESS
000781  03A7  8000              DC    Z'8000'             EVEN BYTE, 250 MS DELAY
000782                *
000783                *
000784  03A8  9380 041A          LNJ   $B1,<CRWB           COMPARE READ AND WRITE BLOCKS
000785  03AA  C870 0300          LDR   $R4,=768            NUMBER OF BYTES TO CCB AREA
000786  03AC  CF00 14D7          STR   $R4,<RANGEW
000787                *
000788  03AE  BE00 14CD          SWR   $R3,<TPR            GET WRITE SUBCH.
000789  03B0  9380 1069          LNJ   $B1,<SDATA          WRITE DATA PATTERN
000790  03B2  1EBB              DC    <SDB
000791  03B3  0600              DC    1536
000792  03B4  0800              DC    2048                RAM ADDRESS
000793  03B5  8000              DC    Z'8000'             EVEN BYYTE,250 MS DELAY
000794                *
000795                *
000796  03B6  9380 0411          LNJ   $B1,<SET            FILL READ AREA WITH SET PATTERN
000797                *
000798  03B8  BE00 14CD          SWR   $R3,<TPR            GET READ SUBCH.
000799  03BA  9380 1098          LNJ   $B1,<RDATA          READ DATA PATTERN FROM RAM
000800  03BC  2BAA              DC    <RTB
000801  03BD  0600              DC    1536
000802  03BE  0800              DC    2048                RAM ADDRESS
000803  03BF  8000              DC    Z'8000'             EVEN BYTE, 250 MS DELAY
000804                *
000805                *
000806  03C0  9380 041A          LNJ   $B1,<CRWB           COMPARE READ AND WRITE BLOCKS
000807                *
000808                *
000809                *
000810                *
000811                *  FIND RAM LOCATION AND NUMBER OF BYTES FOR
000812                *     BLOCK READ AND WRITE
000813                *
000814  03C2  C853              LDR   $R4,=$R3            COPY SUBCH.
000815  03C3  4F20              MLV   $R4,=X'20'
000816  03C4  CF00 03FF          STR   $R4,<Q5             DISTANCE FROM E00 TO CHANNEL CCP AREA
000817  03C6  CF00 0409          STR   $R4,<Q6             FOR BLOCK READ AND WRITE
000818  03C8  CA70 1240          ADD   $R4,=X'1240'
000819  03CA  CF00 03E2          STR   $R4,<Q2             ADDRESS ABOVE CCBS IN USE
000820  03CC  CF00 03EC          STR   $R4,<Q4             FOR BLOCK READ AND WRITE
000821  03CE  C270 1280          SUB   $R4,=X'1280'
000822  03D0  8254              NEG   =$R4
000823  03D1  CF00 03E1          STR   $R4,<Q1             STORE NUMBER OF BYTES FOR BLOCK
000824  03D3  CF00 03EB          STR   $R4,<Q3             READ AND WRITE BELOW CURRENT CHANNEL
000825  03D5  4041              SOR   $R4,1               FIND AND STORE NUMBER OF WORDS
000826  03D6  CF00 14D7          STR   $R4,<RANGEW         FOR BLOCK COMPARE
000827                *
000828  03D8  BE00 14CD          SWR   $R3,<TPR            GET WRITE SUBCH.
000829  03DA  9800 03E1          LDR   $R1,<Q1             GET RANGE
000830  03DC  1A80 03F5          BLEZ  $R1,<MTM10          SKIP NEXT TEST FOR LAST LINE
000831                *
000832  03DE  9380 1069          LNJ   $B1,<SDATA          WRITE DATA TO RAM
000833  03E0  1EBB              DC    <SDB
000834  03E1  0000      Q1      DC    0                   RANGE = TOP - CCB TOP
000835  03E2  0000      Q2      DC    0                   RAM ADDRESS, CCB TOP
000836  03E3  8000              DC    Z'8000'             EVEN BYTE, 250 MS DELAY
000837                *
000838                *
000839                *
000840  03E4  9380 0411          LNJ   $B1,<SET            FILL READ BACK AREA WITH SET PATTERN
000841                *
000842  03E6  BE00 14CD          SWR   $R3,<TPR            GET READ SUBCH.
000843  03E8  9380 1098          LNJ   $B1,<RDATA          READ DATA PATTERN FROM RAM
000844  03EA  2BAA              DC    <RTB
000845  03EB  0000      Q3      DC    0                   RANGE = TOP - CCB TOP
000846  03EC  0000      Q4      DC    0                   RAM ADDRESS, CCB TOP
000847  03ED  8000              DC    Z'8000'             EVEN BYTE, 250 MS DELAY
000848                *
000849                *
000850  03EE  9380 041A          LNJ   $B1,<CRWB           COMPARE READ AND WRITE BLOCKS
000851                *
000852  03F0  D870 1240          LDR   $R5,=X'1240'        SEE IF YOU HAVE TO TEST
000853  03F2  D900 03E2          CMR   $R5,<Q2             BELOW CURRENT CCB
000854  03F4  091A              BE    >TESM1
000855                *
000856                *  TEST MEMORY FROM E00 TO CCB IN USE
000857                *
000858  03F5  C800 03FF  MTM10   LDR   $R4,<Q5             FIND RANGE IN WORDS FOR
000859  03F7  4041              SOR   $R4,1               BLOCK COMPARE
000860  03F8  CF00 14D7          STR   $R4,<RANGEW
000861                *
000862  03FA  BE00 14CD          SWR   $R3,<TPR            GET WRITE SUBCH.
000863  03FC  9380 1069          LNJ   $B1,<SDATA          SEND DATA PATTERN TO RAM
000864  03FE  1EBB              DC    <SDB
000865  03FF  0000      Q5      DC    0
000866  0400  1200              DC    X'1200'
000867  0401  8000              DC    Z'8000'             EVEN BYTE, 250 MS DELAY
000868                *
000869                *
000870  0402  9380 0411          LNJ   $B1,<SET            FILL READ BACK AREA WITH SET PATTERN
000871                *
000872  0404  BE00 14CD          SWR   $R3,<TPR            GET READ SUBCH.
000873  0406  9380 1098          LNJ   $B1,<RDATA          READ DATA PATTERN FROM RAM
000874  0408  2BAA              DC    <RTB
000875  0409  0000      Q6      DC    0
000876  040A  1200              DC    X'1200'             RAM ADDRESS
000877  040B  8000              DC    Z'8000'             EVEN BYTE, 250 MS DELAY
000878                *
000879                *
000880  040C  9380 041A          LNJ   $B1,<CRWB           COMPARE READ AND WRITE BLOCKS
```

```
000881                                          *
000882  040E  B800 14CF                         TESM1  LDR    $R3,<TEMP1          GET TESTED CHANNEL
000883  0410  8386                                    JMP    $B6                 RETURN FROM SUBROUTINE
000884                                          *
000885                                          *
000886                                          SET    CALL   ZV$F,RTB,DFLT,RANGEW
        0411  FBC0 0003
        0413  D380 0000       X
        0415  0F80
        0416  2BAA
        0417  14BE
        0418  14D7
000887  0419  8381                                    JMP    $B1
000888                                          *
000889                                          *
000890                                          *      COMPARE SEND BLOCK WITH RETURN BLOCK
000891                                          *
000892                                          CRWB   CALL   ZV$C,SDB,RTB,MASK,RANGEW,ERAR
        041A  FBC0 0003
        041C  D380 0000       X
        041E  0F80
        041F  1EBB
        0420  2BAA
        0421  14E1
        0422  14D7
        0423  14E3
000893  0424  89C0 10BE                               CMZ    ERAR                SEE IF ERROR PRESENT
000894  0426  090F                                    BE     >MBM                IF PRESENT HALT WITH
000895  0427  C800 14D8                               LDR    $R4,<RAMAD          STARTING LOCATION OF BLOCK IN RAM
000896  0429  D800 14E5                               LDR    $R5,<ERAR+2         'IS' IN $R5
000897  042B  E800 14E4                               LDR    $R6,<ERAR+1         'SHOULD BE' IN $R6
000898  042D  F800 14E3                               LDR    $R7,<ERAR           LOCATION NUMBER OF BLOCK IN $R7
000899  042F  F380 125C                               LNJ    $B7,<ERROR          RAM MEMORY ERROR
000900                                                CALL   ZV$CO               CONTINUE TESTING FOR ERRORS
        0431  FBF0 0001
        0433  D380 0000       X
000901  0435  8381                             MBM    JMP    $B1
000902                                          *-------------------------------------------------------------
000903                                          *
000904                                          *
000905                                          *
000906                                          *
000907                                          *        UNAVAILABLE RESOURCE TEST  ******* DELETED ***************
000908                                          *
000909                                          *
000910  0436  B870 4420                         MNKT   LDR    $R3,=A'D '
000911  0438  BF00 1581                               STR    $R3,<ERMG+1
000912                                                CALL   ZV$T.ZV$TC,TSTD
        043A  FBC0 0003
        043C  D380 0000       X
        043E  0F80
        043F  15A9
000913  0440  8753                                    CL     =$R3                INITIALIZE CHANNEL
000914  0441  C380 10FC                               LNJ    $B4,<GENITZ         INITIALIZE
000915                                          *
000916  0443  C380 10EC                               LNJ    $B4,<FLN            FIND LINE ON
000917  0445  8382                                    JMP    $B2                 NO LINE ON JUMP OUT OF TEST
000918  0446  8751                                    CL     =$R1                CLEAR INDEX FOR OUTPUT ADDRESS
000919  0447  5C20                                    LDV    $R5,=32             LOAD RANGE WITH NUMBER OF BYTES
000920  0448  DF00 14D5                               STR    $R5,<RANGE
000921                                                CALL   ZV$RD.ZV$HM         PUT MEMORY HIGH INTO $B7
        044A  FBF0 0001
        044C  D380 0000       X
000922  044E  C800 155C                               LDR    $R4,<CONT1          FORM I/O CONTROL WORD
000923  0450  C380 1173                               LNJ    $B4,<CGSCH
000924  0452  8187                                    IOLD   $B7,=$R4,=$R5
        0453  0054
        0454  0055
000925  0455  0703                                    BIOT   >$+2+$AF
000926  0456  F380 125C                               LNJ    $B7,<ERROR          I/O WAS NAK'ED
000927  0458  C800 155E                               LDR    $R4,<CONT3          FORM I/O CONTROL WORD
000928  045A  C380 1173                               LNJ    $B4,<CGSCH
000929  045C  8070 0037                               IO     =55,=$R4
        045E  0054
000930  045F  0703                                    BIOT   >$+2+$AF
000931  0460  F380 125C                               LNJ    $B7,<ERROR
000932                                          *
000933  0462  C380 1177                               LNJ    $B4,<CHCT           DO CHANNEL CONTROL
000934  0464  0F82                                    B      >$+2
000935  0465  0400                                    DC     2'0400'             BLOCK WRITE
000936                                          *
000937  0466  C380 1205                               LNJ    $B4,<DLAY           TIME DELAY
000938                                          *
000939  0468  C380 1161                               LNJ    $B4,<INXT           INPUT NEXT STATUS
000940                                          *
000941  046A  82D5                                    LB     =$R5,=2'0001'       SEE IF STATUS BIT SET
        046B  0001
000942  046C  0503                                    BBT    >IS64K
000943  046D  F380 125C                               LNJ    $B7,<ERROR          NON EXISTANT RESOURCE BIT NOT SET
000944  046F  8382                             IS64K  JMP    $B2                 JUMP OUT OF TEST
000945                                          *
000946                                          *-------------------------------------------------------------
000947                                          *
000948                                          *
000949                                          *        CHECK OUTPUT BYTE INTO LCT AND
000950                                          *         CHECK INPUT BYTE COMMAND
000951                                          *
000952                                          *
000953                                          *
000954  0470  B870 4520                         LCTOB  LDR    $R3,=A'E '
000955  0472  BF00 1581                               STR    $R3,<ERMG+1
000956                                                CALL   ZV$T.ZV$TC,TSTE
        0474  FBC0 0003
        0476  D380 0000       X
        0478  0F80
        0479  15B7
000957  047A  C380 10FC                               LNJ    $B4,<GENITZ         GIVE GENERAL INITIALIZE
000958                                          *
000959  047C  8753                                    CL     =$R3
000960  047D  C380 10EC                         LCTB   LNJ    $B4,<FLN            FIND LINE ON
000961  047F  8382                                    JMP    $B2                 NO LINE ON
000962  0480  BF00 0538                               STR    $R3,<TSTCHN         STORE SUBCH.
000963  0482  3B00 0485                               BEVN   $R3,<NXLCTA         SET TO READ SUBCH.
```

```
000964  0484  88D3                      DEC     =$R3
000965                                  *
000966                                  *  SET UP TABLE TO BE SENT IN OUTPUT BYTE TO LCT
000967                                  *
000968  0485  C380 0489                 NXLCTA LNJ     $B4,<LCTGRB          FILL LCT AREA WITH A PATTERN
000969  0487  0F80 04CD                        B       <LCTZ
000970                                  *
000971                                  *  DATA IS ASCENDING STARTING WITH ONE
000972                                  *
000973  0489  5CC0                      LCTGRB LDV     $R5,=-64             64 ENTRIES IN TABLE
000974  048A  8751                            CL      =$R1                 R1 TRACKS ADDRESS
000975  048B  A870 0100                        LDR     $R2,=X'100'          R2 TRACKS DATA
000976  048D  C852                     NXLCT  LDR     $R4,=$R2
000977  048E  C451                            OR      $R4,=$R1             FORM DATA AND ADDRESS
000978  048F  CF10 1EBB                        STR     $R4,<SDB.$R1
000979  0491  8AD1                            INC     =$R1
000980  0492  AA70 0100                        ADD     $R2,=X'100'          FORM NEXT DATA WORD
000981  0494  5780 048D                        BINC    $R5,<NXLCT           DO NEXT VALUE
000982                                  * NOW DUMMY OUT CONTROL CHARACTERS
000983  0496  C870 0302                        LDR     $R4,=Z'0302'
000984  0498  8751                            CL      =$R1
000985  0499  A810 04AC                NXLCT1 LDR     $R2,<NONO.$R1        GET FORBIDDEN CHARACTER
000986  049B  8AD1                            INC     =$R1
000987  049C  2800 04A1                        BLZ     $R2,<GRB1            BRANCH IF END OF LIST
000988  049E  CF20 1EBB                        STR     $R4,<SDB.$R2
000989  04A0  0FF9                            B       >NXLCT1
000990                                  *
000991                                  * SET DATA BUFFER POINTERS TO NON ZERO
000992                                  *
000993  04A1  9870 8005                 GRB1   LDR     $R1,=Z'8005'
000994  04A3  9F00 1EC0                        STR     $R1,<SDB+5
000995  04A5  9870 4025                        LDR     $R1,=Z'4025'
000996  04A7  9F00 1ED4                        STR     $R1,<SDB+25
000997  04A9  8700 1EFB                        CL      <SDB+64
000998  04AB  8384                            JMP     $B4                  GET OUT
000999                                  *
001000                                  * TABLE OF CONTROL LOCATIONS
001001                                  *
001002  04AC  0008                      NONO   DC      X'8'                 CHANNEL COMMAND REC
001003  04AD  0009                             DC      X'9'                 CHANNEL CONTROL REC
001004  04AE  0013                             DC      X'13'
001005  04AF  0021                             DC      X'21'
001006  04B0  0022                             DC      X'22'
001007  04B1  0023                             DC      X'23'
001008  04B2  0028                             DC      X'28'                CHAN COMMAND, TRAN
001009  04B3  0029                             DC      X'29'                CHAN CONTROL, TRAN
001010  04B4  000D                             DC      X'D'
001011  04B5  002D                             DC      X'2D'
001012  04B6  0020                             DC      X'20'
001013  04B7  0035                             DC      X'35'
001014  04B8  0000                             DC      X'0'
001015  04B9  0016                             DC      X'16'
001016  04BA  0014                             DC      X'14'                LR CONTROL
001017  04BB  0015                             DC      X'15'
001018  04BC  0032                             DC      X'32'
001019  04BD  0040                             DC      X'40'
001020  04BE  0041                             DC      X'41'
001021  04BF  0045                             DC      X'45'
001022  04C0  0053                             DC      X'53'
001023  04C1  0055                             DC      X'55'
001024  04C2  0036                             DC      X'36'
001025  04C3  0037                             DC      X'37'
001026  04C4  FFFF                             DC      -X'1'                END OF LIST
001027  04C5  0000                             RESV    8,0                  ROOM FOR EXPANSION
001028                                  *
001029                                  *
001030  04C6  9870 0017                 LCTZ   LDR     $R1,=23
001031  04C8  C870 0017                        LDR     $R4,=23              START SENDING FROM LCT 23
001032  04CA  9B80 1EBB                        LAB     $B1,<SDB             SEND BUFFER
001033  04CC  CF52                      LCTCR  STR     $R4,=$R2
001034  04CD  2008                            SOL     $R2,8
001035  04CE  A454                            OR      $R2,=$R4             SET BOTH ADDRESS OR DATA
001036  04CF  AF5D                            STR     $R2,$B1.+$R1         CREAT LCT BUFFER
001037  04D0  8AD4                            INC     =$R4
001038  04D1  9970 0020                       CMR     $R1,=32              IS IT LCT 32
001039  04D3  0981 FFF8                       BNE     LCTCR                CREAT MORE, IF NOT
001040  04D5  8752                            CL      =$R2
001041  04D6  AF11                            STR     $R2,$B1.$R1          MARK END OF THE BUFFER
001042  04D7  B380 0FE0                       LNJ     $B3,<SNDLCT          SENT LCT BYTES TO RAM
001043  04D9  1EBB                            DC      <SDB                 BUFFER ADDRESS
001044  04DA  0016                            DC      22
001045  04DB  A870 0017                       LDR     $R2,=23
001046  04DD  BB80 1EBB                       LAB     $B3,<SDB
001047  04DF  9823                     RBYTES LDR     $R1,$B3.$R2
001048  04E0  9F56                            STR     $R1,=$R6
001049  04E1  9570 FF00                       AND     $R1,=Z'FF00'         ALIGN ADDRESS
001050  04E3  9470 0037                       OR      $R1,=X'0037'         IN BYTE DIRECTOR
001051  04E5  C380 0E0C                       LNJ     $B4,<OUTLCT          OUTPUT BYTE
001052  04E7  C380 11A8                       LNJ     $B4,<INBYTE          INPUT LCT BYTE
001053  04E9  E570 FF00                       AND     $R6,=Z'FF00'         STRIPT TO DATA
001054  04EB  D956                            CMR     $R5,=$R6
001055  04EC  0901 0003                       BE      RBYTE1               BRANCH IF DATA IS RIGHT
001056  04EE  F380 125C                       LNJ     $B7,<ERROR           OR ELSE REPOT ERROR
001057  04F0  8AD2                     RBYTE1 INC     =$R2
001058  04F1  A970 0020                       CMR     $R2,=32
001059  04F3  0981 FFEB                       BNE     RBYTES
001060  04F5  C800 0538                       LDR     $R4,<TSTCHN
001061  04F7  4B80 0505                       BODD    $R4,<LCTE
001062  04F9  4F20                            MLV     $R4,=X'0020'         CALCUTLE LCT RAM ADDRESS
001063  04FA  CA70 0017                       ADD     $R4,=23              START FROM LCT 23
001064  04FC  CF00 0509                       STR     $R4,<LCTA
001065  04FE  9380 1098                LCTE   LNJ     $B1,<RDATA           READ LCT 23 TO 31
001066  0500  2BAA                            DC      <RTB                 READ BUFFER
001067  0501  0009                            DC      9
001068  0502  0000                     LCTA   DC      0
001069  0503  8000                            DC      =Z'8000'
001070                                  *
001071                                  *
001072                                  *          PACK THE UNPACKED DATA
001073                                  *
001074  0504  2CF8                            LDV     $R2,=-8
001075  0505  BB80 1EDA                       LAB     $B3,<SDB+31
001076  0507  C823                     LCTC   LDR     $R4,$B3.$R2
```

```
001077  050F  C570 00FF            AND     $R4,=Z'00FF'
001078  0511  CF23                 STR     $R4,$B3,$R2
001079  0512  27FC                 BINC    $R2,>LCTC
001080  0513  5C08                 LDV     $R5,=8
001081  0514  C380 11B6            LNJ     $B4,<PKRIN
001082  0516  1ED2                 DC      <SDB+23
001083  0517  1F1F                 DC      <SDB+100
001084                        *
001085                        *
001086                        *
001087                        *  COMPARE THE BLOCK READ DATA
001088                        *
001089  0518  8751                 CL      =$R1
001090  0519  BB80 1F1F            LAB     $B3,<SDB+100
001091  051B  9B80 2BAA            LAB     $B1,<RTB
001092  051D  D813              LCTX  LDR     $R5,$B3,$R1
001093  051E  E811                 LDR     $R6,$B1,$R1
001094  051F  D956                 CMR     $R5,=$R6
001095  0520  0903                 BE      >$+2+$AF
001096  0521  F380 125C            LNJ     $B7,<ERROR
001097  0523  8AD1                 INC     =$R1
001098  0524  9970 0003            CMR     $R1,=3
001099  0526  0981 FFF6            BNE     LCTX
001100                        *
001101                        *EECHECK TO SEE IF LCT STATUS IS CLEARED
001102                        *                       (DONE FOR RECEIVE CHANNELS ONLY)
001103                        *
001104  0528  3B80 0531            BODD    $R3,<NOER5
001105  052A  C380 084E            LNJ     $B4,<ILCTST             READ LCT STATUS
001106                        *
001107                        *
001108  052C  8756                 CL      =$R6                    LOAD $R6 WITH 'SHOULD BE'
001109  052D  D956                 CMR     $R5,=$R6
001110  052E  0903                 BE      >$+2+$AF
001111                        *
001112  052F  F380 125C            LNJ     $B7,<ERROR              LCT STATUS NOT CLEARED
001113                        *
001114  0531  B800 0538         NOER5  LDR     $R3,<TSTCHN             GET HIGHEST SUBCH.
001115  0533  8AD3                 INC     =$R3
001116  0534  3D10                 CMV     $R3,=16
001117  0535  0981 FF47            BNE     LCTB
001118  0537  8382                 JMP     $B2
001119  0538  0000             TSTCHN DC      0
001120                        *
001121                        *-------------------------------------------------------------
001122                        *
001123                        *
001124                        * SOFT-INITIALIZE TEST
001125                        *
001126  0539  9870 5349            SITEST LDR     $R1,=A'SI'              REPORT TEST
001127  053B  9F00 1581                 STR     $R1,<ERMG+1
001128                                  CALL    ZV$T,ZV$TC,TSTSI
          053D  FBC0 0003
          053F  D380 0000       X
          0541  0F80
          0542  15C5
001129                        *
001130  0543  C380 10FC            SIT1   LNJ     $B4,<GENITZ             INITIALIZE
001131                        *
001132  0545  8753                 CL      =$R3
001133  0546  C380 10EC            LNJ     $B4,<FLN                FIND LINE NUMBER
001134  0548  8382                 JMP     $B2                     NO LINES
001135                        *
001136                        * FILL WRITE BUFFER WITH DATA (8408)
001137                        *
001138                                  CALL    ZV$F,SDB,CCITT,BUFRNG
          0549  FBC0 0003
          054B  D380 0000       X
          054D  0F80
          054E  1EBB
          054F  14A5
          0550  14C6
001139                        *
001140                        * WRITE DATA OUT
001141  0551  BF00 14CD            SIT9   STR     $R3,<TPR
001142  0553  3B80 0556                 BODD    $R3,<SIT4
001143  0555  8AD3                 INC     =$R3                    GET WRITE CHANNEL
001144                        *
001145  0556  9380 1069            SIT4   LNJ     $B1,<SDATA
001146  0558  1EBB                 DC      <SDB                    FROM HERE
001147  0559  0400                 DC      X'400'                  RANGE
001148  055A  0300                 DC      X'300'                  START AT HEX 300
001149  055B  8000                 DC      Z'8000'                 EVEN BYTE BOUNDRY
001150                        *
001151                        * WRITE OUT CHANNEL PROGRAM
001152                        *
001153  055C  9380 1069            LNJ     $B1,<SDATA              SEND DATA
001154  055E  05AA                 DC      <ITEST
001155  055F  0008                 DC      (ITEST1-ITEST)*2        RANGE
001156  0560  0200                 DC      X'200'                  RAM ADDRESS
001157  0561  0000                 DC      0                       EVEN BYTE
001158                        *
001159                        * LOAD DUMMY INFO TO LCT 8, X'28'
001160                        *
001161  0562  B380 0FCC            LNJ     $B3,<SETLCT
001162  0564  05A5                 DC      <IZLCT
001163                        *
001164                        * READ LCT 8,9,X'28',X'29'
001165                        *
001166  0565  B800 14CD            LDR     $R3,<TPR                GET TESTED CHANNEL
001167  0567  B380 0628            LNJ     $B3,<STRTIO             START CHANNEL PROGRAM
001168  0569  C380 1205            LNJ     $B4,<DLAY               DELAY 25 MS
001169  056B  9B80 05AE            SIT2   LAB     $B1,<TSVLU
001170  056D  8752                 CL      =$R2
001171  056E  986D                 SIT6   LDR     $R1,$B1,+$R2            GET BYTE ADDRESS
001172  056F  C380 0E0C            LNJ     $B4,<OUTLCT             OUTPUT II
001173  0571  C380 11A8            LNJ     $B4,<INBYTE             INPUT BYTE
001174  0573  5903                 BEZ     $R5,>SIT5               BRANCH IF CLEARED
001175  0574  F380 125C            LNJ     $B7,<ERROR              LCT NOT CLEARED
001176  0576  2D04                 SIT5   CMV     $R2,=4                  CHECK FOR LAST IN TABLE
001177  0577  09F7                 BNE     >SIT6
001178                        *
001179                        * TEST SOFTWARE SOFT INITIALIZE
```

```
001180                                  *
001181                                  *
001182  0578  C380 0E25                         LNJ     $B4,<RPVLU              READ P VALUE
001183  057A  E870 0204                         LDR     $R6,=X'204'             WHAT IT SHOULD BE
001184  057C  D956                             CMR     $R5,=$R6
001185  057D  0900 0581                         BE      <SIT7
001186  057F  F380 125C                         LNJ     $B7,<ERROR              CCP SOFT INIT FAILURE
001187                                  *
001188  0581  B800 14CD                 SIT7    LDR     $R3,<TPR
001189  0583  3B00 0586                         BEVN    $R3,<SIT11
001190  0585  88D3                             DEC     =$R3
001191                                  *
001192  0586  9380 1098                 SIT11   LNJ     $B1,<RDATA              READ DATA
001193  0588  2BAA                             DC      <RTB                    TO HERE
001194  0589  0400                             DC      X'400'                  RANGE
001195  058A  0300                             DC      X'300'                  RAM ADDRESS
001196  058B  8000                             DC      Z'8000'                 EVEN BYTE, 250 MS DELAY
001197                                  *
001198  058C  8700 14E1                         CL      <MASK
001199  058E  9870 0200                         LDR     $R1,=X'200'
001200  0590  9F00 14D7                         STR     $R1,<RANGEW             RANGE IN WORDS
001201  0592  9380 041A                         LNJ     $B1,<CRWB               COMPARE DATA
001202  0594  C380 1177                 SIT10   LNJ     $B4,<CHCT               CHANNEL CONTROL
001203  0596  0F82                             B       >$+2
001204  0597  0100                             DC      X'100'                  CCB LIST RESET
001205                                  *
001206  0598  8AD3                             INC     =$R3                    GET WRITE CHANNEL
001207  0599  C380 1177                         LNJ     $B4,<CHCT               CHANNEL CONTROL
001208  059B  0F82                             B       >$+2
001209  059C  0100                             DC      X'100'                  CCB LIST RESET
001210                                  *
001211                                  *
001212  059D  B800 14CD                         LDR     $R3,<TPR                GET TESTED CHANNEL
001213  059F  8AD3                             INC     =$R3                    BUMP FOR NEXT CHANNEL
001214  05A0  C380 10EC                         LNJ     $B4,<FLN                FIND NEXT LINE NUMBER
001215  05A2  8382                             JMP     $B2                     NO MORE LINES, DONE
001216  05A3  0F80 0551                         B       <SIT9
001217                                  *
001218                                  *
001219                                  * LCT FOR THIS TEST
001220  05A5  4008                      IZLCT   DC      Z'4008'                 DATA SET SCAN
001221  05A6  4028                              DC      Z'4028'
001222  05A7  0206                      IZL     DC      Z'0206'                 P, MSB RCV
001223  05A8  0226                              DC      Z'0226'                 P MSB, XMIT
001224  05A9  0000                              DC      0
001225                                  *
001226                                  * CHANNEL PROGRAM
001227                                  *
001228  05AA  01BD                      ITEST   DC      Z'01BD'                 NOP,INZ
001229  05AB  31A0                              DC      Z'31A0'                 NOP,WAIT
001230  05AC  01BD                              DC      Z'01BD'
001231  05AD  3100                              DC      Z'3100'
001232        05AE                      ITEST1  EQU     $
001233                                  *
001234                                  * TEST ADDRESSES
001235                                  *
001236  05AE  0837                      TSVLU   DC      Z'0837'
001237  05AF  0937                              DC      Z'0937'
001238  05B0  2837                              DC      Z'2837'
001239  05B1  2937                              DC      Z'2937'
001240                                  *
001241                                  * CHANNEL PROGRAMS FOR NEXT TEST
001242                                  *
001243                                  *
001244  05B2  01BD                      PROG1   DC      Z'01BD'                 NOP, WAIT
001245  05B3  3100                              DC      Z'3100'
001246  05B4  01BD                              DC      Z'01BD'                 NOP, WAIT
001247  05B5  3100                              DC      Z'3100'
001248  05B6  20FE                              DC      Z'20FE'                 B -2
001249                                  *
001250                                  *----------------------------------------------------------------
001251                                  *
001252                                  *
001253                                  *          CHANNEL PROGRAM TEST (NOP PROGRAM)
001254                                  *
001255                                  *
001256  05B7  B870 4620                 CHPGT   LDR     $R3,=A'F '
001257  05B9  BF00 1581                         STR     $R3,<ERMG+1
001258                                          CALL    ZV$I,ZV$TC,TSIF
        05BB  FBC0 0003
        05BD  D380 0000         X
        05BF  0F80
        05C0  15D0
001259  05C1  8753                             CL      =$R3                    INITIALIZE CHANNEL NUMBER
001260  05C2  C380 10FC                         LNJ     $B4,<GENITZ             GENERAL INITIALIZE
001261                                  *
001262  05C4  C380 10EC                         LNJ     $B4,<FLN                FIND LINE ON
001263  05C6  8382                             JMP     $B2                     NO LINE ON JUMP OUT OF TEST
001264  05C7  3B80 05CA                         BODD    $R3,<CHPGT2             GET WRITE SUBCH.
001265  05C9  8AD3                             INC     =$R3
001266  05CA  8751                      CHPGT2  CL      =$R1                    CLEAR INDEX FOR OUTPUT ADDRESS
001267  05CB  BF00 14CD                         STR     $R3,<TPR                STORE SUBCH.
001268  05CD  E870 0800                         LDR     $R6,=Z'0800'            NOP BUFFER RANGE
001269  05CF  EF40 0EF0                         STR     $R6,RNG                 STORE AWAY
001270                                          CALL    ZV$F,SDB,PNOP,RNG
        05D1  FBC0 0003
        05D3  D380 0000         X
        05D5  0F80
        05D6  1EBB
        05D7  14BF
        05D8  14C0
001271  05D9  9380 1069                         LNJ     $B1,<SDATA              SEND PATTERN TO RAM
001272  05DB  1EBB                             DC      <SDB                    BUFFER ADDRESS
001273  05DC  1000                             DC      X'1000'                 RANGE
001274  05DD  0200                             DC      X'200'                  STARTING ADDRESS OF RAM
001275  05DE  8000                             DC      Z'8000'
001276                                  *
001277                                  *
001278  05DF  9380 1069                         LNJ     $B1,<SDATA              WRITE CHANNEL PROGRAM
001279  05E1  05B2                             DC      <PROG1
001280  05E2  000A                             DC      X'000A'                 RANGE
001281  05E3  11F6                             DC      X'11F6'
001282  05E4  0000                             DC      0                       EVEN BYTE BOUNDRY
```

```
001283                                  *
001284                                  *
001285  05E5  B380 0FCC                         LNJ     $B3,<SETLCT             SEND LCT OUT FOR THIS CHANNEL
001286  05E7  0637                             DC      <CPT4
001287                                  *
001288  05E8  C380 1205                         LNJ     $B4,<DLAY
001289  05EA  B380 0628                         LNJ     $B3,<STRTIO             START IO
001290                                  *
001291                                  *
001292  05EC  E870 0200                         LDR     $R6,=Z'0200'
001293  05EE  CF00 14D4                         STR     $R4,<IRG
001294  05F0  9870 FF20                         LDR     $R1,=-224
001295  05F2  0F03                      NP4    NOP     >$+2+$AF
001296  05F3  1780 05F2                         BINC    $R1,<NP4                SHORT DELAY FOR CHAN PROG TO START.
001297  05F5  C380 1177                         LNJ     $B4,<CHCT
001298  05F7  0F82                              B       >$+2
001299  05F8  2000                              DC      Z'2000'                 STOP IO
001300                                  *
001301  05F9  C380 1205                         LNJ     $B4,<DLAY               DELAY 25MS
001302  05FB  C380 0E25                         LNJ     $B4,<RPVLU              READ P TO TEMP
001303                                  *
001304  05FD  D956                              CMR     $R5,=$R6                R6 STILL HAS X'221'
001305  05FE  0303                              BG      >NP5
001306  05FF  F380 125C                         LNJ     $B7,<ERROR              PROGRAM DIDN'T START AGAIN
001307                                  *
001308  0601  E870 1200                 NP5    LDR     $R6,=X'1200'
001309  0603  D956                              CMR     $R5,=$R6
001310  0604  0203                              BL      >NP6
001311  0605  F380 125C                         LNJ     $B7,<ERROR              ERROR, STOP I/O DIDN'T STOP  CCP
001312                                  * START CHANNEL PROGRAM AGAIN
001313                                  *
001314                                  * RESET P TO START AT LOC WHICH RESETS START I/O BIT
001315                                  *
001316  0607  B380 0FCC                 NP6    LNJ     $B3,<SETLCT
001317  0609  0636                             DC      <CPT5
001318  060A  B380 0628                         LNJ     $B3,<STRTIO             GIVE START IO
001319  060C  1CB0                              LDV     $R1,=-80
001320  060D  0F03                      NP7    NOP     >$+2+$AF
001321  060E  1780 060D                         BINC    $R1,<NP7                DELAY IO LET CHANNEL PROGRAM START
001322                                  *
001323                                  * GIVE STOP I/O FOR OTHER CHANNEL.  SHOULDN'T STOP PROGRAM.
001324                                  *
001325  0610  88D3                              DEC     =$R3
001326  0611  C380 1177                         LNJ     $B4,<CHCT               DO CHANNEL CONTROL
001327  0613  0F82                              B       >$+2
001328  0614  2000                              DC      Z'2000'                 STOP IO
001329                                  *
001330                                  *
001331                                  * GIVE OUTPUT BYTE COMMANDS TO MAKE SURE THEY DON'T STOP CHANNEL PROG.
001332                                  *
001333  0615  8AD3                              INC     =$R3                    GET BACK CHANNEL
001334  0616  B380 0FCC                         LNJ     $B3,<SETLCT             OUTPUT LCT AGAIN
001335  0618  0633                             DC      <CPT2                   SAME AS BEFORE
001336                                  *
001337                                  * NOW GIVE INPUT DATA SET STATUS TO CHECK IT WONT TERMINATE CHAN
001338                                  * PROGRAM
001339                                  *
001340  0619  C380 0EC6                         LNJ     $B4,<DSSTA
001341  061B  C380 1205                         LNJ     $B4,<DLAY               DELAY 25 MS
001342                                  *
001343  061D  C380 0E25                         LNJ     $B4,<RPVLU              READ P TO TEMP
001344                                  *
001345                                  *
001346  061F  E870 11FA                         LDR     $R6,=Z'11FA'            FORM 'SHOULD BE' P COUNTER
001347  0621  CF00 14D4                         STR     $R4,<IRG                STORE FOR ERROR
001348  0623  D956                              CMR     $R5,=$R6                COMPARE P COUNTER
001349  0624  0903                              BE      >$+2+$AF
001350  0625  F380 125C                         LNJ     $B7,<ERROR              P COUNT WRONG, BAD PROGRAM EXICUTION
001351  0627  8382                              JMP     $B2                     DONE WITH TEST
001352                                  *
001353                                  * START IO
001354                                  *
001355  0628  C800 1562                 STRTIO LDR     $R4,<CONT7              GET FUNC CODE
001356  062A  C380 1173                         LNJ     $B4,<CGSCH
001357  062C  8070 4000                         IO      =Z'4000',=$R4
       062E  0054
001358  062F  0703                              BIOT    >$+2+$AF
001359  0630  F380 125C                         LNJ     $B7,<ERROR              COMMAND WAS NAK'ED
001360  0632  8383                              JMP     $B3
001361                                  *
001362  0633  0105                      CPT2   DC      Z'0105'                 RCV PAUSE DISABLE
001363  0634  0125                             DC      Z'0125'                 XMIT PAUSE DISABLE
001364  0635  0000                             DC      0
001365                                  *
001366  0636  2827                      CPT5   DC      X'2827'                 RESTART AT 22a
001367  0637  0226                      CPT4   DC      X'0226'                 P,MSB,XMIT
001368  0638  0000                             DC      0                       END OF LIST
001369                                  *
001370                                  *-------------------------------------------------------------
001371                                  *
001372                                  *           INSTRUCTION TEST
001373                                  *
001374                                  *
001375  0639  B870 4720                 INSTT  LDR     $R3,=A'G '
001376  063B  BF00 1581                         STR     $R3,<ERMG+1
001377                                          CALL    ZV$T.ZV$TC,TSTG
       063D  FBC0 0003
       063F  D380 0000          X
       0641  0F80
       0642  15DB
001378  0643  8753                             CL      =$R3                    INITIALIZE CHANNEL NUMBER
001379                                  *
001380  0644  C380 10EC                 INSLUP LNJ     $B4,<FLN                FIND LINE ON
001381  0646  8382                              JMP     $B2                     NO LINE ON JUMP OUT OF TEST
001382  0647  BF00 14CD                         STR     $R3,<TPK                STORE SUBCH. NUMBER
001383  0649  BF00 14CF                         STR     $R3,<TEMP1
001384  064B  C380 10FC                         LNJ     $B4,<GENITZ             MLCC GENERAL INITIALIZE
001385  064D  3B82                              BUDD    $R3,>INS2               GET WRITE SUBCH.
001386  064E  8AD3                              INC     =$R3
001387  064F  8751                      INS2   CL      =$R1                    CLEAR INDEX FOR OUTPUT ADDRESS
001388  0650  C870 0364                         LDR     $R4,=X'0364'            LOAD MASK1 WITH EXPECTED
001389  0652  CF00 14E2                         STR     $R4,<MASK1              P COUNTER
001390                                  *
```

```
001391                                  *
001392                                  *
001393                                  *       PREPARE AND LOAD RAM INSTRUCTIONS
001394                                  *
001395                                  *
001396  0654  9380 1069                        LNJ     $B1,<SDATA
001397  0656  1291                            DC      <INSTR1
001398  0657  00EE                            DC      (INSTR2-INSTR1)*2
001399  0658  0200                            DC      X'200'                  RAM ADDRESS
001400  0659  0000                            DC      0                       CPU HAS RIGHT BYTE START
001401                                  *
001402                                  *
001403  065A  9380 1069                        LNJ     $B1,<SDATA
001404  065C  1308                            DC      <INSTR2
001405  065D  0076                            DC      (INSTR3-INSTR2)*2
001406  065E  02EE                            DC      X'02EE'                 RAM ADDRESS
001407  065F  0000                            DC      0                       EVEN BYTE START
001408                                  *
001409                                  *
001410  0660  9380 1069                        LNJ     $B1,<SDATA
001411  0662  1343                            DC      <INSTR3
001412  0663  0060                            DC      (INSTR4-INSTR3)*2
001413  0664  036A                            DC      X'36A'                  RAM ADDRESS
001414  0665  0000                            DC      0                       EVEN BYTE START
001415                                  *
001416  0666  9380 1069                        LNJ     $B1,<SDATA              SEND DATA
001417  0668  1373                            DC      <INSTR4
001418  0669  0008                            DC      (INSTR5-INSTR4)*2       RANGE
001419  066A  03C8                            DC      X'3C8'                  RAM ADDRESS
001420  066B  0000                            DC      0                       EVEN BYTE IN CPU
001421                                  *
001422  066C  9380 1069                        LNJ     $B1,<SDATA              SEND DATA
001423  066E  1377                            DC      <INSTR5
001424  066F  000A                            DC      (INSTR6-INSTR5)*2       RANGE
001425  0670  03D0                            DC      X'3D0'                  RAM ADDRESS
001426  0671  0000                            DC      0                       EVEN BYTE IN CPU
001427                                  *
001428  0672  9380 1069                        LNJ     $B1,<SDATA              SEND DATA
001429  0674  137C                            DC      <INSTR6
001430  0675  000A                            DC      (TRANM1-INSTR6)*2       RANGE
001431  0676  03DA                            DC      X'3DA'                  RAM ADDRESS
001432  0677  0000                            DC      0                       EVEN BYTE IN CPU
001433                                  *
001434                                  *
001435                                  * SEND OUT LOOKUP TABLE FOR INSTRUCTION TEST
001436                                  *
001437  0678  9380 1069                        LNJ     $B1,<SDATA              SEND PROGRAM
001438  067A  06B3                            DC      <TALU
001439  067B  0004                            DC      4                       RANGE OF 4 BYTES
001440  067C  0402                            DC      X'402'                  RAM ADDRESS
001441  067D  0000                            DC      0                       STARTS ON EVEN BOUNDRY
001442                                  *
001443                                  *
001444                                  *
001445  067E  B380 0FCC                        LNJ     $B3,<SETLCT             SEND LCT OUT
001446  0680  06A5                            DC      <LCTTAB
001447  0681  4C01                            LDV     $R4,=1
001448  0682  C380 1189                        LNJ     $B4,<MCCB               DUMMY CCB TO MAKE LAST CHARA FALSE
001449  0684  2BAA                            DC      <RTB
001450  0685  0040                            DC      X'40'
001451  0686  B380 0628                        LNJ     $B3,<STRTIO             START IO
001452                                  *
001453                                  *
001454  0688  C380 1215                        LNJ     $B4,<DLAYLG             TIME DELAY
001455                                  *
001456  068A  C853                            LDR     $R4,=$R3                FIND RAM ADDRESS TO READ P COUNTER
001457  068B  4F20                            MLV     $R4,=X'20'
001458  068C  4E06                            ADV     $R4,=6
001459  068D  CF00 0694                        STR     $R4,<INS1
001460  068F  88D3                            DEC     =$R3                    GET READ SUBCH.
001461                                  *
001462  0690  9380 1098                        LNJ     $B1,<RDATA              READ P COUNTER BACK
001463  0692  14D7                            DC      <RANGEW
001464  0693  0002                            DC      2
001465  0694  0000                    INS1    DC      0
001466  0695  8000                            DC      2'8000'                 500 MS. DELAY
001467                                  *
001468                                  *
001469  0696  D800 14D7                        LDR     $R5,<RANGEW
001470  0698  D570 0FFF                        AND     $R5,=X'FFF'
001471  069A  E800 14E2                        LDR     $R6,<MASK1              GET SHOULD BE DATA
001472  069C  D956                            CMR     $R5,=$R6                SEE IF P COUNTER RIGHT?
001473  069D  0903                            BE      >NOINF                  IF NOT EQUAL HALT
001474  069E  F380 125C                        LNJ     $B7,<ERROR              INSTRUCTION TEST ERROR $R5 CONTAINS PCOUNTER
001475  06A0  B800 14CD                NOINF   LDR     $R3,<TPR                GET BACK TESTED CHANNEL
001476  06A2  8AD3                            INC     =$R3                    BUMP CHANNEL NUMBER
001477  06A3  0F81 FFA0                        B       INSLUP                  DO FOR NEXT CHANNEL
001478                                  *
001479                                  * LCT TABLE FOR THIS TEST
001480                                  *
001481  06A5  0226                    LCTTAB  DC      2'0226'                 LCT CCP POINTER (MSB)
001482  06A6  EE27                            DC      2'EE27'                 CCP POINTER (LSB)
001483  06A7  FF03                            DC      2'FF03'                 ADDRESS TESTING VALUE
001484  06A8  AA1C                            DC      2'AA1C'                 DITTO
001485  06A9  552C                            DC      2'552C'                 DITTO
001486  06AA  AA3F                            DC      2'AA3F'                 DITTO
001487  06AB  0437                            DC      2'0437'                 TLU POINTER, MSB
001488  06AC  0238                            DC      2'0238'                 TLU POINTER, LSB
001489  06AD  0105                            DC      2'0105'
001490  06AE  0125                            DC      2'0125'
001491  06AF  FF17                            DC      2'FF17'
001492  06B0  0000                            DC      0
001493  06B1  0000                            DC      0
001494  06B2  0000                            DC      0                       MARKS END OF TABLE
001495                                  *
001496                                  * TABLE FOR TABLE LOOKUP TEST
001497                                  *
001498  06B3  031A                    TALU    DC      X'031A'
001499  06B4  8500                            DC      2'8500'
001500                                  *-----------------------------------------------------------------
001501                                  *
001502                                  * OUTPUT INTERRUPT CONTROL TEST
001503                                  *
```

```
001504  06B5  B870 4820          INCONT LDR    $R3,=A'H '
001505  06B7  BF00 1581                 STR    $R3,<ERMG+1
001506                                  CALL   ZV$T,ZV$TC,TSTH
        06B9  FBC0 0003
        06BB  D380 0000      X
        06BD  0F80
        06BE  15E4
001507  06BF  8753                     CL     =$R3                 R3 TRACKS CHANNEL
001508  06C0  C380 10EC          ICON3  LNJ    $B4,<FLN             FIND ACTIVE LINE NUMBER
001509  06C2  8382                     JMP    $B2                  NO MORE LINES
001510  06C3  8C56                     STS    =$R6                 GET S REG
001511  06C4  E570 0FC0                 AND    $R6,=X'FC0'          STRIP IO RETURN CHANNEL NUMBER
001512  06C6  6E3F                     ADV    $R6,=63              SET  FOR LEVEL 63
001513  06C7  C380 10FC                 LNJ    $B4,<GENITZ          GENERAL INITIALIZE
001514  06C9  C800 1565                 LDR    $R4,<CONT11          OUTPUT INTERRUPT CONTROL  FUNCTION
001515  06CB  C380 1173                 LNJ    $B4,<CGSCH           MODIFY FOR CHANNEL
001516  06CD  8056                     IO     =$R6,=$R4            OUTPUT INTERRUPT CONTROL
        06CE  0054
001517  06CF  0703                     BIOT   >$+2+$AF
001518  06D0  F380 125C                LNJ    $B7,<ERROR           COMMAND WAS NAK'ED
001519                           *
001520  06D2  C853                     LDR    $R4,=$R3
001521  06D3  4F20                     MLV    $R4,=X'20'           OFFSET FOR CHANNEL
001522  06D4  6E0C                     ADV    $R4,=X'C'            OFFSET FOR RETURN CHANNEL
001523  06D5  CF00 06E0                STR    $R4,<ICON2
001524  06D7  BF00 14CD                STR    $R3,<TPR
001525  06D9  3B00 06DC                BEVN   $R3,<ICON1
001526  06DB  88D3                     DEC    =$R3                 GET RCV CHANNEL FOR BLOCK READ
001527                           *
001528  06DC  9380 1098          ICON1  LNJ    $B1,<RDATA
001529  06DE  14CE                     DC     <TEMP                READ TO HERE
001530  06DF  0002                     DC     2                    2 BYTES OF DATA
001531  06E0  0000               ICON2  DC     0                    RAM  ADDRESS
001532  06E1  0000                     DC     0                    EVEN BYTE ADDRESS
001533                           *
001534  06E2  B800 14CD                LDR    $R3,<TPR             RESTORE R3
001535  06E4  D800 14CE                LDR    $R5,<TEMP
001536  06E6  D956                     CMR    $R5,=$R6             CHECK IS VS SB
001537  06E7  0903                     BE     >$+2+$AF
001538  06E8  F380 125C                LNJ    $B7,<ERROR           ERROR, OUTPUT INTERRUPT CONTROL
001539  06EA  8AD3                     INC    =$R3                 SET FOR NEXT CHANNEL
001540  06EB  0F81 FFD4                B      ICON3                DO NEXT CHAN
001541                           *
001542                           *-------------------------------------------------------------
001543                           *
001544                           * INITIALIZE AND CHANNEL RESET TEST
001545                           *
001546                           *   FIRST GIVE GEN INITIALIZE + CHECK IT CLEARS LCT AREA
001547                           *
001548  06ED  B870 4A20          IZTST  LDR    $R3,=A'J '
001549  06EF  BF00 1581                 STR    $R3,<ERMG+1
001550                                  CALL   ZV$T,ZV$TC,TSTJ
        06F1  FBC0 0003
        06F3  D380 0000      X
        06F5  0F80
        06F6  15F4
001551  06F7  8753                     CL     =$R3                 R3 TRACKS CHANNEL
001552  06F8  C380 10EC          IZTST6 LNJ    $B4,<FLN             FIND ACTIVE LINES
001553  06FA  8382                     JMP    $B2                  NO MORE ACTIVE LINES
001554                           *
001555  06FB  C380 0489                 LNJ    $B4,<LCTGRB          SEND GARBAGE TO LCT AREA
001556  06FD  C800 155C                 LDR    $R4,<CONT1           IOLD FUNCTION CODE
001557  06FF  C380 1173                 LNJ    $B4,<CGSCH           MODIFY FOR CHANNEL
001558  0701  CF52                     STR    $R4,=$R2
001559  0702  C800 1563                 LDR    $R4,<CONT9           INITIALIZE FUNCTION CODE
001560  0704  C380 1173                 LNJ    $B4,<CGSCH
001561  0706  8070 8000                 IO     =Z'8000',=$R4        INITIALIZE
        0708  0054
001562  0709  0703                     BIOT   >$+2+$AF
001563  070A  F380 125C                LNJ    $B7,<ERROR           INITIALIZE WAS NAK'ED
001564                           *
001565  070C  81C0 0DC1                 IOLD   TEMP,=$R2,=X'1'      GIVE ANOTHER INSTRUCTION
        070E  0052
        070F  0070 0001
001566  0711  0783                     BIOF   >$+2+$AF
001567  0712  F380 125C                LNJ    $B7,<ERROR           SHOULD BE NAK'ED UNTIL INITIALIZE DONE
001568                           *
001569  0714  8070 8000                 IO     =Z'8000',=$R4        GIVE ANOTHER INITIALIZE
        0716  0054
001570  0717  0783                     BIOF   >$+2+$AF
001571  0718  F380 125C                LNJ    $B7,<ERROR           INITIALIZE DID NOT NAK'ED
001572                           *
001573  071A  C380 1215                 LNJ    $B4,<DLAYLG          DELAY 25 MS **********
001574                           * SET DEFAULT VALUE IN READ BUFFER
001575  071C  9870 0100                 LDR    $R1,=X'100'
001576  071E  9F00 14D5                 STR    $R1,<RANGE
001577                                  CALL   ZV$F,RTB,DFLT,RANGE
        0720  FBC0 0003
        0722  D380 0000      X
        0724  0F80
        0725  2BAA
        0726  14BE
        0727  14D5
001578                           * READ IN LCT AREA FOR THIS CHANNEL AND CHECK
001579  0728  C380 07C3                 LNJ    $B4,<BRLCLR          CHECK LCT IS CLEAR
001580  072A  B800 14CD                 LDR    $R3,<TPR             RESTORE CHAN NUMBER
001581                           *-------------------------------------------------------------
001582                           *
001583                           * CHANNEL INITIALIZE TEST
001584                           *
001585                           *   SEND OUT  GARBAGE TO LCT AREA
001586                           *
001587  072C  C380 0489                 LNJ    $B4,<LCTGRB
001588  072E  C870 0302                 LDR    $R4,=Z'0302'
001589  0730  CF00 1ECB                 STR    $R4,<SDB+16
001590  0732  CF00 1ECC                 STR    $R4,<SDB+17
001591  0734  CF00 1EC0                 STR    $R4,<SDB+5
001592  0736  CF00 1EEB                 STR    $R4,<SDB+48
001593  0738  CF00 1EEC                 STR    $R4,<SDB+49
001594  073A  CF00 1EE0                 STR    $R4,<SDB+37
001595  073C  B380 0FE0                 LNJ    $B3,<SNDLCT          SEND OUT LCT BYTES
001596  073E  1EBB                     DC     <SDB
001597  073F  FFFF                     DC     -1
```

```
001598                                  *
001599                                  * NOW SET UP 4 CCB'S TO INSURE INITIALIZE CLEARS  IOLD POINTER
001600  0740  9380 080C                        LNJ     $B1,<SET4
001601                                  * ADVANCE INPUT NEXT STAT POINTER A COUPLE OF TIMES
001602  0742  9380 07F1                        LNJ     $B1,<ADV2
001603                                  * NOW GIVE CHANNEL INITIALIZE
001604  0744  C800 1562                        LDR     $R4,<CONT7              CHANNEL CONTROL
001605  0746  C380 1173                        LNJ     $B4,<CGSCH              MODIFY FOR CHANNEL
001606  0748  8070 8000                        IO      =Z'8000',=$R4
        074A  0054
001607  074B  0703                             BIOT    >$+2+$AF
001608  074C  F380 125C                        LNJ     $B7,<ERROR              OUTPUT CHAN CONT NAK'ED
001609  074E  C380 1205                        LNJ     $B4,<DLAY               DELAY 25 MS
001610                                  * USE OTHER SIDE OF LINE TO BLOCK READ CCB AREA + CHECK
001611  0750  BF00 14CD                        STR     $R3,<TPR
001612  0752  3F20                             MLV     $R3,=X'20'
001613  0753  BA70 0E00                        ADD     $R3,=Z'0E00'
001614  0755  BF00 075F                        STR     $R3,<CADD               ADDRESS OF START OF CCB AREA
001615  0757  B800 14CD                        LDR     $R3,<TPR
001616  0759  8B53                             LBC     =$R3,=Z'0001'           GET OTHER SIDE OF LINE
        075A  0001
001617                                  *
001618  075B  9380 1098                        LNJ     $B1,<RDATA              BLOCK READ
001619  075D  2BAA                             DC      <RTB                    RETURN BUFFER
001620  075E  0020                             DC      X'20'                   20 BYTE RANGE
001621  075F  0000                      CADD   DC      0                       ADDRESS
001622  0760  0000                             DC      0                       EVEN BYTE BOUNDRY
001623                                  *
001624                                  * CHECK CONTROL BYTES WERE CLEARED
001625  0761  1CFC                             LDV     $R1,=-4
001626  0762  2C05                             LDV     $R2,=5
001627  0763  D0A0 2BAA                 LUP    LDM     $R5,<RTB,$R2            GET CONTROL BYTE
001628  0765  5903                             BEZ     $R5,>LUP1               BRANCH IF GOOD
001629  0766  F380 125C                        LNJ     $B7,<ERROR              CHAN INIT. DIDN'T RESET DATA POINTERS IN LR 5
001630  0768  2E08                      LUP1   ADV     $R2,=8                  SET FOR NEXT CONTROL BYTE
001631  0769  17FA                             BINC    $R1,>LUP                DO FOR NEXT
001632  076A  B800 14CD                        LDR     $R3,<TPR                GET BACK TESTED SUBCHANNEL
001633                                  *
001634                                  * CHECK INPUT NEXT STATUS WILL BE NAK'ED AFTER CHANNEL INITIALIZE
001635                                  *
001636  076C  C800 155F                        LDR     $R4,<CONT4
001637  076E  C380 1173                        LNJ     $B4,<CGSCH
001638  0770  8055                             IO      =$R5,=$R4               GIVE INPUT NEXT STATUS COMMAND
        0771  0054
001639  0772  0783                             BIOF    >$+2+$AF                SHOULD BE NAK'ED
001640  0773  F380 125C                        LNJ     $B7,<ERROR              COMMAND WASN'T NAK'ED
001641  0775  0F07                             NOP     >ECO2
001642                                  *
001643                                  * SET UP 4 CCB'S  AGAIN TO CHECK LOAD POINTER WAS RESET
001644  0776  9380 080C                 ECO3   LNJ     $B1,<SET4
001645  0778  9380 07F6                        LNJ     $B1,<CHKRNG             CHECK STATUS POINTER IS CORRECT
001646  077A  9380 07F1                        LNJ     $B1,<ADV2               ADVANCE POINTER 2
001647                                  *
001648                                  * TEST RESET CCB LIST COMMAND
001649                                  *
001650  077C  0F81 0002                 ECO2   B       ECO4                    SET DATA BUFF POINTER'S NON ZERO
001651  077E  07C0                             DC      <DTPT
001652  077F  C800 1562                 ECO4   LDR     $R4,<CONT7              CHANNEL CONTROL
001653  0781  C380 1173                        LNJ     $B4,<CGSCH              MODIFY FOR  CHANNEL
001654  0783  8070 0100                        IO      =Z'0100',=$R4           RESET CCB LIST
        0785  0054
001655  0786  0703                             BIOT    >$+2+$AF
001656  0787  F380 125C                        LNJ     $B7,<ERROR              COMMAND WAS NAK'ED
001657                                  * SET UP 4 CCB'S AGAIN TO CHECK IF CCB RESET WORKS
001658                                  *
001659  0789  9380 080C                        LNJ     $B1,<SET4               SET UP 4 CCB'S
001660  078B  9380 07F6                        LNJ     $B1,<CHKRNG             CHECK INPUT STAT POINTER  IS RESET
001661                                  *
001662                                  * NOW CHECK CHANNEL ECT AREA WAS CLEARED BY INITIALIZE
001663                                  *
001664  078D  C380 1177                        LNJ     $B4,<CHCT
001665  078F  0F82                             B       >$+2
001666  0790  0100                             DC      X'100'                  RESET CCB LIST
001667                                  *
001668                                  *
001669                                  *     SET UP DEFAULT
001670                                  *
001671  0791  4C45                             LDV     $R4,=X'45'
001672  0792  CF00 14D7                        STR     $R4,<RANGEW
001673  0794  9380 0411                        LNJ     $B1,<SET
001674  0796  C380 07C3                        LNJ     $B4,<BRLCLR             CHECK LCT AREA FOR CHAN IS CLEAR
001675                                  *
001676                                  * TEST BVBT + BVBF COMMANDS
001677                                  *
001678  0798  C380 10FC                        LNJ     $B4,<GENITZ             GENERAL INITIALIZE
001679                                  *
001680  079A  9380 1069                        LNJ     $B1,<SDATA              SEND CHANNEL PROGRAM
001681  079C  147D                             DC      <VALTST                 ADDRESS
001682  079D  004E                             DC      (ECP-VALTST)*2          RANGE
001683  079E  0200                             DC      X'200'                  RAM ADDRESS
001684  079F  0000                             DC      0                       EVEN CP ADDRESS
001685                                  *
001686                                  * SET UP RCV, XMIT FOR P START AT 200
001687                                  *
001688  07A0  B380 0FCC                        LNJ     $B3,<SETLCT
001689  07A2  05A7                             DC      <IZL
001690                                  *
001691                                  * SEND OUT 4 CCB'S WITH VALID BITS
001692  07A3  9380 080C                        LNJ     $B1,<SET4
001693                                  *
001694  07A5  B380 0628                        LNJ     $B3,<STRTIO             START CHANNEL PROGRAM
001695  07A7  C380 1205                        LNJ     $B4,<DLAY
001696  07A9  C380 1177                        LNJ     $B4,<CHCT               CHANNEL CONTROL
001697  07AB  0F82                             B       >$+2
001698  07AC  0100                             DC      X'100'                  RESET CCB LIST
001699                                  *
001700                                  * START SECOND HALF OF CHANNEL PROGRAM
001701                                  *
001702  07AD  B380 0628                        LNJ     $B3,<STRTIO
001703  07AF  C380 1205                        LNJ     $B4,<DLAY
001704                                  * READ P TO INSURE CORRECT CHAN PROGRAM OPERATION
001705  07B1  C380 1177                        LNJ     $B4,<CHCT               CHANNEL CONTROL
001706  07B3  0F82                             B       >$+2
```

```
001707  07B4  0100                    DC      X'100'                  RESET CCB LIST
001708  07B5  C380 0E25                LNJ     $B4,<RPVLU              READ P
001709  07B7  E870 024C                LDR     $R6,=X'24C'             **************
001710  07B9  0956                    CMR     $R5,=$R6
001711  07BA  0903                    BE      >$+2+$AF
001712  07BB  F380 125C                LNJ     $B7,<ERROR              BVBI, BVBF, OR CCB LIST RESET ERROR
001713  07BD  8AD3                    INC     =$R3                    BUMP TO NEXT CHANNEL
001714  07BE  0F81 FF39                B       IZTST6
001715                          *
001716  07C0  C005              DTPT    DC      Z'C005'                 RCV DATA PTR
001717  07C1  C025                      DC      Z'C025'                 XMIT DATA POINTER
001718  07C2  0000                      DC      0                       END OF LIST
001719                          *---------------------------------------------------------------
001720                          *
001721                          *SUBROUTINES FOR INITIALIZE TEST
001722                          *
001723                          *
001724                          *     BLOCK READ A CHANNEL'S LCT AREA TO CHECK IT'S CLEAR
001725                          *
001726  07C3  C853              BRLCLR  LDR     $R4,=$R3
001727  07C4  4F20                      MLV     $R4,X'20'               FORM RAM ADDRESS OF LCT AREA
001728  07C5  CF00 07D0                 STR     $R4,<IZTST4
001729  07C7  BF00 14CD                 STR     $R3,<TPR
001730  07C9  3B00 07CC                 BEVN    $R3,<IZTST2
001731  07CB  88D3                      DEC     =$R3
001732                          *
001733  07CC  9380 1098         IZTST2  LNJ     $B1,<RDATA              READ RAM
001734  07CE  2BAA                      DC      <RTB                    RECEIVE BUFFER
001735  07CF  0020                      DC      X'20'
001736  07D0  0000              IZTST4  DC      0                       RAM ADDRESS
001737  07D1  0000                      DC      0                       START ON EVEN
001738                          *
001739  07D2  B800 14CD                 LDR     $R3,<TPR
001740                          *
001741                          * CHECK ZERO'S WERE READ IN
001742                          *
001743  07D4  8756                      CL      =$R6
001744  07D5  2C01                      LDV     $R2,=1                  START WITH WORD 2
001745  07D6  D820 2BAA         IZTST1  LDR     $R5,<RTB.$R2            GET LCT CONTENT
001746  07D8  0956                      CMR     $R5,=$R6                IS LCT CONTENT=0
001747  07D9  0903                      BE      >$+2+$AF                CONTINUE IF ZERO
001748  07DA  F380 125C                 LNJ     $B7,<ERROR              OR ELSE REPORT ERROR
001749  07DC  8AD2              IZTST7  INC     =$R2                    INCREMENT COUNTER
001750  07DD  AF57                      STR     $R2,=$R7
001751  07DE  3B90                      BODD    $R3,>IZTST3             BRANCH IF TRANSMIT CHANNEL
001752  07DF  A970 0002                 CMR     $R2,=X'0002'
001753  07E1  0901 FFFA                 BE      IZTST7
001754  07E3  A970 0004                 CMR     $R2,=X'0004'            IS IT BLOCK READ BIT
001755  07E5  0901 FFF6                 BE      IZTST7
001756  07E7  A970 000A                 CMR     $R2,=X'000A'
001757  07E9  0901 FFF2                 BE      IZTST7
001758  07EB  A970 000B                 CMR     $R2,=X'000B'
001759  07ED  096F                      BE      >IZTST7                 OR CONTROL WORD
001760  07EE  2D10              IZTST3  CMV     $R2,=X'10'
001761  07EF  09E7                      BNE     >IZTST1
001762  07F0  8384                      JMP     $B4                     ALL DONE
001763                          *
001764                          *                                       .
001765                          *
001766                          * DO TWO INPUT NEXT STATUS TO GET THE INPUT
001767                          * NEXT STAT POINTER AWAY FROM IT'S INITIAL STATE
001768  07F1  C380 1161         ADV2    LNJ     $B4,<INXT               INPUT NEXT STATUS
001769  07F3  C380 1161                 LNJ     $B4,<INXT               INPUT NEXT STATUS
001770  07F5  8381                      JMP     $B1
001771                          *
001772                          * READ RANGE TO INSURE INPUT NEXT STATUS POINTER IS RESET
001773                          *
001774  07F6  C800 155F         CHKRNG  LDR     $R4,<CONT4              INPUT NEXT STATUS
001775  07F8  C380 1173                 LNJ     $B4,<CGSCH              MODIFY FOR CHANNEL
001776  07FA  8040 0CD3                 IO      TEMP,=$R4               INPUT NEXT TO GET TO FIRST CCB
      07FC  0054
001777  07FD  0703                      BIOT    >$+2+$AF
001778  07FE  F380 125C                 LNJ     $B7,<ERROR
                                * NOW INPUT RANGE, SHOULD BE = 1 IF POINTER WAS RESET
001780  0800  4EF2                      ADV     $R4,=-X'E'
001781  0801  8055                      IO      =$R5,=$R4               INPUT RANGE
      0802  0054
001782  0803  0703                      BIOT    >$+2+$AF
001783  0804  F380 125C                 LNJ     $B7,<ERROR              COMMAND WAS NAKED
001784  0806  6C01                      LDV     $R6,=1
001785  0807  D956                      CMR     $R5,=$R6
001786  0808  0903                      BE      >$+2+$AF
001787  0809  F380 125C                 LNJ     $B7,<ERROR              INPUT STAT POINTER NOT..
                                *                               EPOINTING AT FIRST CCB
001789  080B  8381                      JMP     $B1
001790                          *
001791                          * SET UP 4 CCB'S
001792                          *
001793  080C  4C01              SET4    LDV     $R4,1
001794  080D  C380 1189         SETA    LNJ     $B4,<MCCB               MAKE CCB
001795  080F  2BAA                      DC      <RTB
001796  0810  0040                      DC      X'40'                   VALID CCB BIT
001797  0811  8AD4                      INC     =$R4
001798  0812  4D05                      CMV     $R4,5
001799  0813  09FA                      BNE     >SETA
001800  0814  8381                      JMP     $B1
001801                          *
001802                          *
001803                          *---------------------------------------------------------------
001804                          *
001805                          *               STOP IO
001806                          *
001807                          *---------------------------------------------------------------
001808  0815  B870 4B20           STIORC  LDR     $R3,=A'K '
001809  0817  BF00 1581                 STR     $R3,<ERMG+1
001810                                  CALL    ZV$T.ZV$TC,TSTK
      0819  FBC0 0003
      081B  D380 0000     X
      081D  0F80
      081E  1606
001811  081F  C380 10FC                 LNJ     $B4,<GENITZ
001812  0821  8753                      CL      =$R3                    R3 TRACKS CHANNEL
001813  0822  C380 10EC         RPSIO   LNJ     $B4,<FLN                FIND ACTIVE LINE
```

```
001814  0824  8382                    JMP     $B2                     NO MORE LINES TO TEST
001815                          * SET UP CONDITIONS FOR STOP IO TEST
001816  0825  C380 0489                LNJ     $B4,<LCTGRB             FILL  LCT AREA WITH DATA
001817  0827  B380 0FCC                LNJ     $B3,<SETLCT             SEND OUT LCT BYTES
001818  0829  1E6B                    DC      <SDB
001819  082A  4C01                    LDV     $R4,=1                  RANGE
001820  082B  C380 1189                LNJ     $B4,<MCCB               DUMMY CCB
001821  082D  2BAA                    DC      <RTB
001822  082E  0400                    DC      X'400'                  RAM ADDRESS
001823                          *
001824                          * GIVE STOP I/O
001825                          *
001826  082F  C380 1177                LNJ     $B4,<CHCT               DO CHANNEL CONTROL
001827  0831  0F82                    B       >$+2
001828  0832  2000                    DC      Z'2000'                 STOP IO
001829                          *
001830  0833  C380 1205                LNJ     $B4,<DLAY               DELAY
001831                          *
001832                          * CHECK THAT STOP I/O OPERATED CORRECTLY
001833                          *
001834  0835  C380 1161                LNJ     $B4,<INXT               INPUT NEXT STATUS
001835                          *
001836  0837  E870 1000                LDR     $R6,=X'1000'            STAT COMPLETE BIT
001837  0839  82D5                    LB      =$R5,=X'1000'
      083A  1000
001838  083B  0503                    BBT     >$+2+$AF
001839  083C  F380 125C                LNJ     $B7,<ERROR              STOP IO DIDN'T CAUSE STATUS COMPLETE
001840                          *
001841                          * INPUT LCT STATUS
001842                          *
001843  083E  C380 084E                LNJ     $B4,<ILCTST
001844  0840  8756                    CL      =$R6
001845  0841  D956                    CMR     $R5,=$R6
001846  0842  0903                    BE      >$+2+$AF
001847  0843  F380 125C                LNJ     $B7,<ERROR              STOP IO DIDN'T CLEAR LCT STATUS
001848                          *
001849                          * DO A CHANNEL INITIALIZE IN PREPARATION FOR THE NEXT TEST
001850                          *
001851  0845  C380 1177                LNJ     $B4,<CHCT
001852  0847  0F82                    B       >$+2
001853  0848  0000                    DC      0                       CHANNEL INITIALIZE
001854  0849  C380 1205                LNJ     $B4,<DLAY               TIME DELAY TO ALLOW INITIALIZE TO COMPLETE
001855                          *
001856  084B  8AD3                    INC     =$R3                    SET FOR NEXT CHANNEL NUMBER
001857  084C  0F81 FFD5                B       RPS10
001858                          *----------------------------------------------------------------
001859                          *
001860                          * SUBROUTINES FOR STOP IO
001861                          *
001862                          *
001863                          * INPUT LCT STATUS TO R5
001864                          *
001865  084E  8F40 0CA8        ILCTST  SAVE    SAV2,=Z'AA88'           R4,B4
      0850  AA88
001866  0851  E870 1137                LDR     $R6,=Z'1137'
001867  0853  A870 1037                LDR     $R2,=Z'1037'
001868  0855  3B00 085B                BEVN    $R3,<ILC1               BRANCH IF RCV CHAN
001869  0857  AA70 2000                ADD     $R2,=Z'2000'
001870  0859  EA70 2000                ADD     $R6,=Z'2000'
001871  085B  9852                    ILC1    LDR     $R1,=$R2        GET BYTE ADDRESS OF MSB OF STAT
001872  085C  C380 0E0C                LNJ     $B4,<OUTLCT             OUTPUT BYTE
001873  085E  C380 11A8                LNJ     $B4,<INBYTE             INPUT BYTE TO R5
001874  0860  DF00 14CD                STR     $R5,<TPR
001875  0862  9856                    LDR     $R1,=$R6
001876  0863  C380 0E0C                LNJ     $B4,<OUTLCT             OUTPUT BYTE
001877  0865  C380 11A8                LNJ     $B4,<INBYTE             INPUT LSB OF STAT
001878  0867  5048                    SOR     $R5,8                   MOVE OVER
001879  0868  D400 14CD                OR      $R5,<TPR                FORM WHOLE STAT WORD
001880  086A  8FC0 0C8C                RSTR    SAV2,=Z'AA88'
      086C  AA88
001881  086D  8384                    JMP     $B4                     EXIT SUBROUTINE
001882                          *----------------------------------------------------------------
001883                          *
001884                          *
001885                          *
001886                          * RECEIVE TESTS
001887                          *
001888                          *
001889                          *
001890                          *   1. EVEN BYTE START, EVEN RANGE, EOR TERMINATION
001891                          *
001892  086E  9870 4C31        RC0     LDR     $R1,=A'L1'
001893  0870  9F00 1581                STR     $R1,<ERMG+1
001894                                  CALL    ZV$I,ZV$TC,TSTL1
      0872  FBC0 0003
      0874  D380 0000       X
      0876  0F80
      0877  160E
001895  0878  C380 090A                LNJ     $B4,<RCVTST             RECEIVE TEST
001896  087A  0F86                    B       >RC1                    END RETURN
001897  087B  0000                    DC      0                       EVEN BYTE BOUNDRY
001898  087C  0006                    DC      6                       CCB RANGE
001899  087D  0040                    DC      X'40'                   CCB CONTROL WORD
001900  087E  0AB2                    DC      <LCT2                   LCT TABLE
001901  087F  0006                    DC      6                       NUMBER OF BYTES TO XFER
001902                          *----------------------------------------------------------------
001903                          *
001904                          *   2. ODD START,<EVEN RANGE, EOR TERMINATION
001905                          *
001906  0880  9870 4C32        RC1     LDR     $R1,=A'L2'
001907  0882  9F00 1581                STR     $R1,<ERMG+1
001908  0884  C380 090A                LNJ     $B4,<RCVTST
001909  0886  0F86                    B       >RC2
001910  0887  0001                    DC      1                       ODD BYTE BOUNDRY
001911  0888  0006                    DC      6                       CCB RANGE
001912  0889  0040                    DC      X'40'                   CCB CONTROL
001913  088A  0AB2                    DC      <LCT2                   LCT TABLE
001914  088B  0006                    DC      6                       NUMBER OF BYTES TO XFER
001915                          *
001916                          *----------------------------------------------------------------
001917                          *
001918                          *   3. EVEN START, EVEN RANGE, GNB TERMINATION
001919                          *
```

```
001920  088C  9870 4C33          RC2     LDR     $R1,=A'L3'
001921  088E  9780 1581                  STR     $R1,<ERMG+1
001922  0890  C380 090A                  LNJ     $B4,<RCVTST             RECEIVE TEST
001923  0892  0F86                      B       >RC3                    DONE
001924  0893  0000                      DC      0                       EVEN BYTE
001925  0894  0008                      DC      8                       CCB RANGE
001926  0895  0040                      DC      X'40'                   CCB CONTROL WORD
001927  0896  0AB1                      DC      <LCT2A                  LCT
001928  0897  0006                      DC      6                       NUMBER OF BYTES TO XFER
001929                          *-----------------------------------------------------------
001930                          *
001931                          *   4. EVEN START, ODD RANGE, GNB TERMINATION
001932                          *
001933  0898  9870 4C34          RC3     LDR     $R1,=A'L4'
001934  089A  9F00 1581                  STR     $R1,<ERMG+1
001935  089C  C380 090A                  LNJ     $B4,<RCVTST             RECEIVE TEST
001936  089E  0F86                      B       >RC4                    DONE
001937  089F  0000                      DC      0                       EVEN BYTE
001938  08A0  0005                      DC      5                       CCB RANGE
001939  08A1  0040                      DC      X'40'                   CCB CONTROL WORD
001940  08A2  0AB1                      DC      <LCT2A                  LCT
001941  08A3  0003                      DC      3                       NUMBER OF BYTES TO XFER
001942                          *-----------------------------------------------------------
001943                          *
001944                          * RECEIVE TESTS
001945                          *
001946                          *5. EVEN START, ODD RANGE, EOR TERMINATION
001947                          *
001948  08A4  9870 4C35          RC4     LDR     $R1,=A'L5'
001949  08A6  9F00 1581                  STR     $R1,<ERMG+1
001950  08A8  C380 090A                  LNJ     $B4,<RCVTST             RECEIVE TEST
001951  08AA  0F86                      B       >RC5                    RETURN
001952  08AB  0000                      DC      0                       EVEN BYTE START
001953  08AC  0005                      DC      5                       CCB RANGE
001954  08AD  0040                      DC      X'40'                   CCB CONTROL
001955  08AE  0AB2                      DC      <LCT2
001956  08AF  0005                      DC      5                       NUMBER OF BYTES TO XFER
001957                          *-----------------------------------------------------------
001958                          *
001959                          * CCB OVER RUN TEST
001960                          *
001961  08B0  9870 4C36          RC5     LDR     $R1,=A'L6'
001962  08B2  9F00 1581                  STR     $R1,<ERMG+1
001963  08B4  C380 090A                  LNJ     $B4,<RCVTST
001964  08B6  8382                      JMP     $B2                     EXIT TEST
001965  08B7  0000                      DC      0
001966  08B8  0003                      DC      3                       CCB RANGE
001967  08B9  1040                      DC      X'1040'                 BIT 3 SET FOR FLAG FOR CCB SERVICE ERROR
001968  08BA  0AB2                      DC      <LCT2
001969  08BB  0005                      DC      5                       NUMBER OF BYTES  TO XFER
001970                          *
001971                          *
001972                          **
001973                          *
001974                          *-----------------------------------------------------------
001975                          *
001976                          * TRANSMIT TESTS
001977                          *
001978                          *-----------------------------------------------------------
001979                          *
001980                          *   1. EVEN START, EVEN RANGE, EOR TERM
001981                          *
001982  08BC  9870 4D31          TRAN1   LDR     $R1,=A'M1'
001983  08BE  9F00 1581                  STR     $R1,<ERMG+1
001984                                   CALL    ZV$T.ZV$TC,TSTM1
        08C0  FBC0 0003
        08C2  D380 0000          X
        08C4  0F80
        08C5  1615
001985  08C6  C3C0 00DA                  LNJ     $B4,XTST                TRANSMIT TEST
001986  08C8  0F86                      B       >TRAN2                  RETURN
001987  08C9  0000                      DC      0                       EVEN BYTES
001988  08CA  0006                      DC      6                       CCB RANGE
001989  08CB  0060                      DC      =X'60'                  CCB CONTROL, LAST BLOCK
001990  08CC  0AAB                      DC      <LCT1                   LCT
001991  08CD  0006                      DC      6                       NUMBER OF BYTES TO XFER
001992                          *-----------------------------------------------------------
001993                          *
001994                          *   2. ODD START,<EVEN RANGE
001995                          *
001996                          *
001997  08CE  9870 4D32          TRAN2   LDR     $R1,=A'M2'
001998  08D0  9F00 1581                  STR     $R1,<ERMG+1
001999  08D2  C3C0 00CE                  LNJ     $B4,XTST                TRANSMIT TEST
002000  08D4  0F86                      B       >TRAN3
002001  08D5  0001                      DC      1                       ODD BYTE
002002  08D6  0006                      DC      6                       CCB RANGE
002003  08D7  0060                      DC      =X'60'                  CCB CONTROL WORD, LAST BLOCK, VALID
002004  08D8  0AAB                      DC      <LCT1                   LCT
002005  08D9  0006                      DC      6                       NUMBER OF BYTES TO XFER
002006                          *
002007                          *-----------------------------------------------------------
002008                          *
002009                          *   3. EVEN START, EVEN RANGE, GNB TERMINATION
002010                          *
002011  08DA  9870 4D33          TRAN3   LDR     $R1,=A'M3'
002012  08DC  9F00 1581                  STR     $R1,<ERMG+1
002013  08DE  C3C0 00C2                  LNJ     $B4,XTST                TRANSMIT TEST
002014  08E0  0F86                      B       >TRAN4                  RETURN
002015  08E1  0000                      DC      0                       EVEN BYTE
002016  08E2  0008                      DC      8                       CCB RANGE
002017  08E3  0060                      DC      X'0060'                 LAST BLOCK, VALID
002018  08E4  0AAA                      DC      <LCT1A                  LCT
002019  08E5  0006                      DC      6                       NUMBER OF BYTES TO XFER
002020                          *
002021                          *-----------------------------------------------------------
002022                          *
002023                          *   4. EVEN START, ODD RANGE, GNB TERMINATION
002024                          *
002025  08E6  9870 4D34          TRAN4   LDR     $R1,=A'M4'
002026  08E8  9F00 1581                  STR     $R1,<ERMG+1
002027  08EA  C3C0 00B6                  LNJ     $B4,XTST                TRANSMIT TEST
002028  08EC  0F86                      B       >TRAN5                  RETURN
```

```
002029  08ED  0000                    DC      0
002030  08EE  0008                    DC      8                       CCB RANGE
002031  08EF  0060                    DC      X'60'                   CCB CONTROL, LAST BLOCK
002032  08F0  0AAA                    DC      <LCT1A                  LCT
002033  08F1  0005                    DC      5                       NUMBER OF BYTES TO XFER
002034                          *
002035                          *---------------------------------------------------------------
002036                          *
002037                          *   5. EVEN START, ODD RANGE, EOR TERMINATION
002038                          *
002039  08F2  9870 4D35         TRAN5   LDR     $R1,=A'M5'
002040  08F4  9F00 1581                 STR     $R1,<ERMG+1
002041  08F6  C3C0 00AA                 LNJ     $B4,XTST                TRANSMIT TEST
002042  08F8  0F86                      B       >TRAN6
002043  08F9  0000                      DC      0                       EVEN START
002044  08FA  0005                      DC      5                       RANGE (CCB)
002045  08FB  0060                      DC      X'60'                   CCB CONTROL , LAST  BLOCK
002046  08FC  0AAB                      DC      <LCT1                   LCT
002047  08FD  0005                      DC      5                       NUMBER OF BYTES TO XFER
002048                          *---------------------------------------------------------------
002049                          *
002050                          * CCB UNDER-RUN TEST
002051                          *
002052  08FE  9870 4D36         TRAN6   LDR     $R1,=A'M6'
002053  0900  9F00 1581                 STR     $R1,<ERMG+1
002054  0902  C3C0 009E                 LNJ     $B4,XTST                TRANSMIT
002055  0904  8382                      JMP     $B2                     EXIT TEST
002056  0905  0000                      DC      0                       EVEN  START
002057  0906  0005                      DC      5                       CCB RANGE
002058  0907  1040                      DC      X'1040'                 NOT LAST CCB
002059  0908  0AAB                      DC      <LCT1                   LCT
002060  0909  0006                      DC      6                       XMIT 5
002061                          *
002062                          *---------------------------------------------------------------
002063                          *
002064                          * RECEIVE TEST
002065                          *
002066                          *   LNJ $B4,<RCVTST
002067                          *   B >NEXT
002068                          *   DC A            1= ODD BYTE
002069                          *   DC B            RANGE IN BYTES
002070                          *   DC C            CCB CONTROL WORD
002071                          *   DC D            POINTS TO LCT TABLE
002072                          *   DC E            NUMBER OF BYTES TO XFER
002073                          *
002074  090A  8F40 0C1C         RCVTST  SAVE    SAV4,=X'547C'           R1,3,4, B1,2,3,4,5
      090C  547C
002075  090D  9870 0200                 LDR     $R1,=X'200'             P COUNTER START
002076  090F  9F00 0A5A                 STR     $R1,<RCWR1
002077  0911  9874                      LDR     $R1,+$B4                DUMMY TO INCREMENT B4
002078  0912  9874                      LDR     $R1,+$B4                ODD/EVEN FLAG
002079  0913  9F00 095F                 STR     $R1,<RCV1
002080  0915  9F00 0966                 STR     $R1,<RCV2
002081  0917  9F00 096F                 STR     $R1,<RCV3
002082  0919  9F00 098B                 STR     $R1,<RCV4
002083                          *
002084  091B  9874                      LDR     $R1,+$B4                PICK UP RANGE
002085  091C  9F00 096B                 STR     $R1,<RCV7
002086                          *
002087  091E  9874                      LDR     $R1,+$B4                PICK UP CCB CONTROL WORD
002088  091F  9F00 0970                 STR     $R1,<RCV9
002089                          *
002090  0921  9CF4                      LDB     $B1,+$B4                LCT POINTER
002091  0922  9F80 096E                 STB     $B1,<RCV10
002092                          *
002093  0924  9874                      LDR     $R1,+$B4                PICK UP NUMBER OF BYTES
002094  0925  9F00 098C                 STR     $R1,<RCV8
002095  0927  9F00 095E                 STR     $R1,<RCV5
002096  0929  9F00 0964                 STR     $R1,<RCV6
002097  092B  8700 14CC                 CL      <FLAG
002098  092D  9900 096B                 CMR     $R1,<RCV7               COMPARE RANGE WITHBYTES TO RCV
002099  092F  090C                      BE      >RCV13
002100  0930  0208                      BL      >RCV14
002101  0931  D800 096B                 LDR     $R5,<RCV7
002102  0933  DF00 095E                 STR     $R5,<RCV5
002103  0935  DF00 098C                 STR     $R5,<RCV8
002104  0937  0F84                      B       >RCV13
002105  0938  4C20              RCV14   LDV     $R4,=X'20'              FLAG SET FOR NON ZERO RANGE RESIDUE
002106  0939  CF00 14CC                 STR     $R4,<FLAG
002107  093B  1008              RCV13   SUL     $R1,8                   FORM RANGE
002108  093C  9670 003F                 XOR     $R1,=X'3F'
002109  093E  9F00 0AB2                 STR     $R1,<LCT2
002110                          *
002111  0940  8753                      CL      =$R3
002112  0941  C380 10EC         NXRCV   LNJ     $B4,<FLN                FIND ACTIVE LINE
002113  0943  0F82                      B       >$+2
002114  0944  0F83                      B       >$+2+$AF
002115  0945  0F81 0057                 B       RCEND                   NO MORE LINES
002116  0947  3B03                      BEVN    $R3,>RCV12              BRANCH MEANS RCV CHAN
002117  0948  8AD3                      INC     =$R3
002118  0949  0FF8                      B       >NXRCV                  TRY NEXT
002119  094A  C853              RCV12   LDR     $R4,=$R3
002120  094B  4F20                      MLV     $R4,=X'20'              FORMS LCT START
002121  094C  4E39                      ADV     $R4,=57
002122  094D  CF00 0965                 STR     $R4,<RC11               RAM ADDRESS WHERE DATA GOES
002123  094F  BF00 095D                 STR     $R3,<RCDATA
002124                          *
002125                          * FILL RECEIVE BUFFER WITH DEFAULT
002126                          *
002127                                  CALL    ZV$F,RTB,DFLT,BUFRNG
      0951  FBC0 0003
      0953  D380 0000        X
      0955  0F80
      0956  2BAA
      0957  14BE
      0958  14C6
002128  0959  C380 10FC                 LNJ     $B4,<GENITZ             MLCC GENERAL INITIALIZE
002129                          * FILL SEND BUFFER WITH SB DATA FOR COMPARISON LATER
002130  095B  C380 0AB8                 LNJ     $B4,<FDTA               FILL ASCENDING DATA
002131  095D  0000              RCDATA  DC      0                       STARTING BYTE
002132  095E  0000              RCV5    DC      0                       RANGE IN BYTES
002133  095F  0000              RCV1    DC      0                       1 = START ON RIGHT BYTE
002134                          *
```

```
002135                                  *
002136                                  * BLOCK WRITE DATA TO RAM. DURING RECEIVE.  THIS DATA WILL BE
002137                                  * TRANSFERED TO CPU BY THE CHANNEL PROGRAM.
002138                                  *
002139  0960  8AD3                             INC     =$R3                    GET XMIT HALF OF LINE
002140                                  *
002141  0961  9380 1069                         LNJ     $B1,<SDATA              SEND DATA
002142  0963  1EBB                             DC      <SDD                    FROM HERE
002143  0964  0000                     RCV6    DC      0                       RANGE IN BYTES
002144  0965  0000                     RC11    DC      0                       RAM ADDRESS
002145  0966  0000                     RCV2    DC      0                       1 = START ON RIGHT BYTE
002146                                  *
002147  0967  88D3                             DEC     =$R3                    GET BACK TO RECEIVE
002148                                  *
002149                                  * RECEIVE A BLOCK OF DATA
002150                                  *
002151  0968  C380 0A36                         LNJ     $B4,<RCV
002152  096A  2BAA                             DC      <RTD                    READ TO HERE
002153  096B  0000                     RCV7    DC      0                       RANGE
002154  096C  13D7                             DC      <RCDTC1                 CHANNEL PROGRAM ADDRESS
002155  096D  00A6                             DC      (RCRT1-RCDTC1)*2        CHANNEL PROGRAM RANGE
002156  096E  14C7                     RCV10   DC      <DUMMY                  LCT PARAMETER TABLE
002157  096F  0000                     RCV3    DC      0                       1= START ON RIGHT BYTE
002158  0970  0000                     RCV9    DC      0                       CCB CONTROL WORD
002159                                  *
002160                                  * READ AND  CHECK STATUS
002161                                  *
002162  0971  C380 1161                         LNJ     $B4,<INXT               INPUT NEXT STATUS
002163  0973  E870 1000                         LDR     $R6,=X'1000'            CCB COMPLETE BIT
002164  0975  E600 14CC                         XOR     $R6,<FLAG               NON ZERO RANGE
002165  0977  D956                             CMR     $R5,=$R6                RESIDUE BIT
002166  0978  0903                             BE      >$+2+$AF
002167  0979  F380 125C                       . LNJ     $B7,<ERROR              STATUS ERROR
002168  097B  8756                             CL      =$R6
002169  097C  82C0 FFF3                       . LB      RCV9,=X'1000'           GET FLAG FOR CCB ERROR
        097E  1000
002170  097F  8A56                             LBS     =$R6,=X'800'            SET CCB SERVICE ERROR BIT
        0980  0800
002171                                  *
002172                                  * READ LCT STATUS
002173                                  *
002174  0981  C380 084E                         LNJ     $B4,<ILCTST             READ LCT STATUS
002175  0983  D570 0800                         AND     $R5,=X'800'             STRIP TO SERVICE ERROR
002176  0985  D956                             CMR     $R5,=$R6
002177  0986  0903                             BE      >$+2+$AF
002178  0987  F380 125C                         LNJ     $B7,<ERROR              CCB SERVICE ERROR BIT  IN WRONG STATE
002179                                  *
002180                                  * CHECK DATA
002181                                  *
002182  0989  D380 0A7F                         LNJ     $B5,<CHKBW              CHECK ROUTINE
002183  098B  0000                     RCV4    DC      0                       1 = START ON RIGHT BYTE
002184  098C  0000                     RCV8    DC      0                       RANGE IN BYTES
002185                                  *
002186                                  * CHECK DATA BUFFER POINTER CLEARED TO 0
002187                                  *
002188  098D  9870 0537                         LDR     $R1,=X'0537'
002189  098F  C380 0E0C                         LNJ     $B4,<OUTLC1             OUTPUT BYTE
002190  0991  C380 11A8                         LNJ     $B4,<INBYTE             INPUT BYTE
002191  0993  D570 C000                         AND     $R5,=Z'C000'            STRIP TO DATA PTR
002192  0995  8756                             CL      =$R6
002193  0996  D956                             CMR     $R5,=$R6
002194  0997  0903                             BE      >RCV11
002195  0998  F380 125C                         LNJ     $B7,<ERROR              GNB DIDN'T RESET DATA PTR'S
002196  099A  8AD3                     RCV11   INC     =$R3
002197                                  *
002198  099B  0F81 FFA5                         B       NXRCV                   DO NEXT LINE
002199                                  *
002200  099D  8FC0 0B89                 RCEND   RSTR    SAV4,=X'547C'
        099F  547C
002201  09A0  8384                             JMP     $B4
002202                                  *
002203                                  *-----------------------------------------------------------------
002204                                  * DATA TRANSMIT TEST
002205                                  *
002206                                  *   LNJ $B4,XTST
002207                                  *   B >NEXT         EXIT BRANCH
002208                                  *   DC 0            1 = ODD BYTE
002209                                  *   DC X            RANGE IN BYTES
002210                                  *   DC Y            CCB CONTROL WORD
002211                                  *   DC Z            POINTER TO LCT PARAMETER TABLE
002212                                  *   DC W            NUMBER OF BYTES TO XFER
002213                                  *
002214                                  *
002215  09A1  8F40 0B85                 XTST    SAVE    SAV4,=X'547C'           R1,3,4, B1,2,3,4,5
        09A3  547C
002216  09A4  9874                             LDR     $R1,+$B4                DUMMY TO INCREMENT B4
002217  09A5  9874                             LDR     $R1,+$B4                PICK UP EVEN, ODD FLAG
002218  09A6  9F00 09E4                         STR     $R1,<CHST1              STORE FLAG FOR ODD OR EVEN BYTE
002219  09A8  9F00 09EC                         STR     $R1,<CHST2
002220  09AA  9F00 0A1C                         STR     $R1,<CHST3
002221  09AC  9F00 0A20                         STR     $R1,<CHST4
002222  09AE  9874                             LDR     $R1,+$B4                PICK UP RANGE
002223  09AF  9F00 09E8                         STR     $R1,<CHRG2
002224                                  *
002225  09B1  8752                             CL      =$R2
002226  09B2  9874                             LDR     $R1,+$B4                PICK UP CCB CONTROL WORD
002227  09B3  82D1                             LB      =$R1,X'40'              GET LAST BLOCK BIT
        09B4  0040
002228  09B5  0503                             BBT     >XTST2                  BRANCH IF NO UNDER-RUN
002229  09B6  A870 0800                         LDR     $R2,=X'800'             UNDER-RUN EXPECTED
002230  09B8  9F00 09ED                 XTST2   STR     $R1,<CCBSAV
002231  09BA  AF00 14CC                         STR     $R2,<FLAG               0 MEANS NO CCB UNDER-RUN
002232                                  *
002233  09BC  9CF4                             LDB     $B1,+$B4                PICK UP LCT POINTER
002234  09BD  9F80 09EB                         STB     $B1,<BWLCT
002235                                  *
002236  09BF  9874                             LDR     $R1,+$B4                PICK UP NUMBER OF BYTES
002237  09C0  9F00 0A21                         STR     $R1,<CHRG4              NUMBER OF BYTES TO CHECK
002238  09C2  9900 09E8                         CMR     $R1,<CHRG2
002239  09C4  0383                             BLE     >XTST3
002240  09C5  9800 09E8                         LDR     $R1,<CHRG2
002241  09C7  9F00 0A1A                 XTST3   STR     $R1,<CHRG3
002242  09C9  9F00 09E3                         STR     $R1,<CHRG1
```

```
002243  09CB  9800 0A21          LDR     $R1,<CHRG4
002244  09CD  1008               SOL     $R1,8
002245  09CE  9670 003F          XOR     $R1,=X'3F'
002246  09D0  9F00 0AAB          STR     $R1,<LCT1
002247                      *
002248  09D2  8753               CL      =$R3
002249  09D3  C380 10EC    NXWRT LNJ     $B4,<FLN                FIND ACTIVE LINE
002250  09D5  0F82               B       >$+2
002251  09D6  0F83               B       >$+2+$AF
002252  09D7  0F81 005A          B       BWEND                   NO MORE LINES
002253  09D9  3B83               BODD    $R3,>XTST1              BRANCH MEANS XMIT CHAN
002254  09DA  8AD3               INC     =$R3
002255  09DB  0FF8               B       >NXWRT                  TRY NEXT
002256  09DC  BF00 09E2    XTST1 STR     $R3,<EODATA
002257  09DE  C380 10FC          LNJ     $B4,<GENITZ             GENERAL INITIALIZE
002258                      *
002259  09E0  C380 0AB8          LNJ     $B4,<FDTA               FILL SDB WITH ASCENDING DTA
002260  09E2  0000         EODATA DC      0                       FIRST DATA WORD
002261  09E3  0005         CHRG1  DC      5                       RANGE IN BYTES
002262  09E4  0000         CHST1  DC      0                       1= START ON RIGHT BYTE
002263                      *
002264  09E5  C380 0A74          LNJ     $B4,<WRT
002265  09E7  1EBB               DC      <SDB                    SEND BUFFER
002266  09E8  0005         CHRG2  DC      5                       RANGE
002267  09E9  1381               DC      <TRANM1                 CHANNEL PROGRAM
002268  09EA  00AC               DC      (RCDTC1-TRANM1)*2       RANGE OF XMIT PROGRAM
002269  09EB  14C7         BWLCT  DC      <DUMMY                  LCT PROTOTYPE
002270  09EC  0000         CHST2  DC      0                       0 = EVEN START
002271  09ED  0000         CCBSAV DC      0                       CCB CONTROL WORD STORED HERE
002272                      *
002273                      * READ AND CHECK STATUS
002274                      *
002275  09EE  8756               CL      =$R6
002276  09EF  C380 1161          LNJ     $B4,<INXT               INPUT NEXT STATUS
002277  09F1  82C0 FFFB          LB      CCBSAV,=X'0020'         GET LAST BLOCK BIT
        09F3  0020
002278  09F4  8A56               LBS     =$R6,=Z'0020'           SET CCB LAST BIT TO EXPECTED STATE
        09F5  0020
002279  09F6  8280 09ED          LB      <CCBSAV,=X'1000'
        09F8  1000
002280  09F9  0583               BBF     >NXT1                   BRANCH IF NOT UNDERUN TEST
002281  09FA  D570 1000          AND     $R5,=Z'1000'
002282  09FC  E470 1000    NXT1  OR      $R6,=Z'1000'            SET STAT  COMP. BIT
002283  09FE  D956               CMR     $R5,=$R6                IS VS SB
002284  09FF  0903               BE      >$+2+$AF
002285  0A00  F380 125C          LNJ     $B7,<ERROR              STATUS ERR AFTER XMIT
002286  0A02  82C0 FFEA          LB      CCBSAV,=X'1000'         GET CCB UNDERRUN FLAG
        0A04  1000
002287  0A05  8756               CL      =$R6
002288  0A06  8A56               LBS     =$R6,=X'800'            CCB SERVICE ERROR BIT
        0A07  0800
002289                      ***
002290                      *
002291                      * READ LCT STATUS
002292                      *
002293                      ***
002294  0A08  C380 084E          LNJ     $B4,<ILCTST             INPUT LCT STATUS
002295  0A0A  D570 0800          AND     $R5,=Z'0800'            STRIP TO CCB SERVICE ERROR
002296  0A0C  D956               CMR     $R5,=$R6
002297  0A0D  0903               BE      >$+2+$AF
002298  0A0E  F380 125C          LNJ     $B7,<ERROR              CCB SERVICE ERROR BIT IN WRONG STATE
002299                      *
002300                      * READ DATA BACK
002301                      *
002302  0A10  C853               LDR     $R4,=$R3
002303  0A11  88D4               DEC     =$R4                    GET RCV CHANNEL
002304  0A12  4F20               MLV     $R4,=X'20'
002305  0A13  4E39               ADV     $R4,=57                 FORM START OF DATA IN RAM
002306  0A14  CF00 0A1B          STR     $R4,<EOADD
002307  0A16  88D3               DEC     =$R3                    GET CHN NUMB FOR RCV HALF OF LINE
002308  0A17  9380 1098          LNJ     $B1,<RDATA              DO BLOCK READ
002309  0A19  2BAA               DC      <RTB                    INPUT BUFFER
002310  0A1A  0005         CHRG3  DC      5                       RANGE
002311  0A1B  0000         EOADD  DC      0                       RAM ADDRESS
002312  0A1C  0000         CHST3  DC      0                       =1 IF BUFFER STARTS ON RIGHT BYTE
002313                      *
002314  0A1D  8AD3               INC     =$R3                    GET XMIT CHAN BACK
002315                      *
002316                      * CHECK DATA
002317                      *
002318  0A1E  D380 0A7F          LNJ     $B5,<CHKBW              CHECK ROUTINE
002319  0A20  0000         CHST4  DC      0                       0 = START LEFT BYTE
002320  0A21  0005         CHRG4  DC      5                       5 BYTES
002321                      *
002322                      * CHECK DATA POINTERS ARE RESET
002323                      *
002324  0A22  9870 2537          LDR     $R1,=X'2537'
002325  0A24  C380 0E0C          LNJ     $B4,<OUTLCT             OUT X'25' TO LOC X'37'
002326  0A26  C380 11A8          LNJ     $B4,<INBYTE             INPUT BYTE
002327  0A28  D570 C000          AND     $R5,=Z'C000'            STRIP TO DATA PTR
002328  0A2A  8756               CL      =$R6
002329  0A2B  D956               CMR     $R5,=$R6
002330  0A2C  0903               BE      >NXT2
002331  0A2D  F380 125C          LNJ     $B7,<ERROR              GNB DIDN'T RESET LR25, BITS 0,1, XMIT
002332  0A2F  8AD3         NXT2  INC     =$R3                    SET FOR NEXT CHANNEL
002333  0A30  0F81 FFA2          B       NXWRT
002334                      *
002335  0A32  8FC0 0AF4    BWEND RSTR    SAV4,=X'547C'
        0A34  547C
002336  0A35  8384               JMP     $B4
002337                      *
002338                      *-----------------------------------------------------------------
002339                      *         LNJ $B4,<RCV
002340                      *         DC  A     BUFFER ADDRESS
002341                      *         DC  B     RANGE VALUE
002342                      *         DC  C     CHANNEL PROGRAM ADDRESS
002343                      *         DC  D    RANGE OF CHANNEL PROGRAM
002344                      *         DC LCT    LCT PARAMETER TABLE
002345                      *         DC  E     0,1 = EVEN, ODD BYTE START ADDRESS
002346                      *         DC CCB    CCB CONTROL WORD
002347                      *
002348                      *
002349  0A36  8F40 0AE0    RCV   SAVE    SAV3,=Z'ECB4'           SAVE REGS.
```

```
                0A38  ECB4
002350          0A39  9870 0200           LDR     $R1,=X'200'
002351          0A3B  9F00 0A5A           STR     $R1,<RCWR1
002352          0A3D  ACF4                LDB     $B2,+$B4          LOAD $B2 WITH ADDRESS
002353          0A3E  C874                LDR     $R4,+$B4          LOAD $R4 WITH NUMBER OF BYTES
002354          0A3F  D870 5555           LDR     $R5,=Z'5555'      LOAD $R5 WITH INITIAL PATTERN
002355          0A41  DF02                STR     $R5,$B2           STORE AT TOP AND BOTTOM OF BUFFER
002356          0A42  CF52                STR     $R4,=$R2
002357          0A43  8AD2                INC     =$R2              FIND INDEX TO LAST WORD
002358          0A44  2041                SOR     $R2,1             R2 HAS WORDS + 1
002359          0A45  DF22                STR     $R5,$B2.$R2
002360                                *
002361          0A46  BCF4        WRT1    LDB     $B3,+$B4          LOAD $B3 WITH ADDRESS OF CHANNEL PROG.
002362          0A47  BF80 0A58           STB     $B3,<RCV2M
002363          0A49  AF80 0A67           STB     $B2,<RCV3M
002364          0A4B  9874                LDR     $R1,+$B4          RANGE OF CHANNEL PROGRAM
002365          0A4C  9F00 0A59           STR     $R1,<RCWR
002366          0A4E  BCF4                LDB     $B3,+$B4          PICK UP ADDRESS OF LCT TABLE
002367          0A4F  BF80 0A60           STB     $B3,<RCLCT
002368          0A51  CFD5                STB     $B4,=$B5
002369          0A52  BF00 14CD           STR     $R3,<TPR          FIND WRITE CHANNEL
002370          0A54  3B82                BODD    $R3,>RCV1M
002371          0A55  8AD3                INC     =$R3
002372          0A56  9380 1069   RCV1M   LNJ     $B1,<SDATA        WRITE OUT CHANNNEL PROGRAM
002373          0A58  14C7        RCV2M   DC      <DUMMY            CHAN PROG
002374          0A59  0000        RCWR    DC      0                 RANGE OF CHANNEL PROGRAM
002375          0A5A  0200        RCWR1   DC      X'200'            RAM ADDRESS
002376          0A5B  0000                DC      0                 RIGHT BYTE START
002377                                *
002378          0A5C  B800 14CD           LDR     $R3,<TPR
002379          0A5E  B380 0FCC           LNJ     $B3,<SETLCT       SEND OUT LCT BYTES
002380          0A60  14C7        RCLCT   DC      <DUMMY            LCT TABLE ADDRESS
002381                                *
002382          0A61  9875                LDR     $R1,+$B5          LOAD $R1 WITH INDEX
002383          0A62  A875                LDR     $R2,+$B5          GET CCB CONTROL WORD
002384          0A63  AF00 0A68           STR     $R2,<RCV4M
002385          0A65  C380 1189           LNJ     $B4,<MCCB         FORM CCB FOR NORMAL OPERATION
002386          0A67  14C7        RCV3M   DC      <DUMMY            CPU DATA ADDRESS
002387          0A68  0010        RCV4M   DC      Z'0010'           CCB CONTROL WORD
002388                                *
002389          0A69  89D2                CMZ     =$R2
002390          0A6A  0803                BAL     >DOLA             BRANCH MEANS FLAG SET TO BYPASS START IO
002391          0A6B  B380 0628           LNJ     $B3,<STRTIO
002392                                *
002393          0A6D  C380 1205   DOLA    LNJ     $B4,<DLAY         TIME DELAY
002394                                *
002395          0A6F  CCD5                LDB     $B4,=$B5
002396          0A70  8FC0 0AA6           RSTR    SAV3,=Z'ECB4'
                0A72  ECB4
002397          0A73  8384                JMP     $B4
002398                                *
002399                                *
002400                                * ELNJE$B4,<WRT
002401                                *EDCECPU ADDRESS OF BUFFER
002402                                * EDCERANGE VALUE
002403                                *EDCECPU ADRESS OF CHANNEL PROGRAM
002404                                *         DC        RANGE OF CHANNEL PROGRAM
002405                                *         DC        LCT  PARAMETER TABLE
002406                                *EDCEO OR 1 ODD OR EVEN BYTE BUFFER ADDRESS START
002407                                *         DC        CCB CONTROL WORD.
002408                                *
002409                                *
002410          0A74  8F40 0AA2   WRT     SAVE    SAV3,=Z'ECB4'     SAVE REGS. R1,2,4,5, B2,3,5 I,<M
                0A76  ECB4
002411          0A77  ACF4                LDB     $B2,+$B4          LOAD $B2 WITH CPU ADDRESS OF BUFFER
002412          0A78  C874                LDR     $R4,+$B4          LOAD $R4 WITH NUMBER OF BYTES
002413          0A79  9870 0400           LDR     $R1,=X'400'
002414          0A7B  9F00 0A5A           STR     $R1,<RCWR1        STORE P COUNTER START
002415          0A7D  0F81 FFC8           B       WRT1
002416                                *
002417                                * CHECK SEND VS. RCV BUFFERS ON A BYTE BY BYTE BASIS.
002418                                *
002419                                *   LNJ $B5,<CHKBW
002420                                *   DC 0              1= START ON RIGHT BYTE
002421                                *   DC X               RANGE IN BYTES
002422                                *
002423          0A7F  9875        CHKBW   LDR     $R1,+$B5          SET BYTE INDEX
002424          0A80  A875                LDR     $R2,+$B5
002425          0A81  8252                NEG     =$R2              RANGE
002426          0A82  BB80 1EBB           LAB     $B3,<SDB          ADDRESS OF SEND BUFFER
002427          0A84  EB80 2BAA           LAB     $B6,<RTB          ADDRESS OF RECEIVE BUFFER
002428                                *
002429          0A86  D096        CHK1    LDH     $R5,$B6.$R1
002430          0A87  E0DF                LDH     $R6,$B3.+$R1      GET SENT DATA
002431          0A88  D1D6                CMH     $R5,=$R6
002432          0A89  0903                BE      >$+2+$AF
002433          0A8A  F380 125C           LNJ     $B7,<ERROR        DATA ERROR
002434          0A8C  2780 0A86           BINC    $R2,<CHK1
002435                                *
002436                                * DATA IS GOOD, CHECK 1 BYTE MORE TO SEE IF INPUT EXCEEDED RANGE
002437                                *
002438          0A8E  D096                LDH     $R5,$B6.$R1
002439          0A8F  6C55                LDV     $R6,=X'55'        WHAT SHOULD BE IN NEXT BYTE
002440          0A90  E955                CMR     $R6,=$R5
002441          0A91  0903                BE      >$+2+$AF
002442          0A92  F380 125C           LNJ     $B7,<ERROR        INPUT EXCEEDED RANGE
002443                                *
002444                                * READ BACK P COUNTER VALUE
002445                                *
002446          0A94  C380 0E25           LNJ     $B4,<RPVLU        READ P VALUE
002447                                *
002448          0A96  B800 14CD           LDR     $R3,<TPR
002449          0A98  3B00 0AA3           BEVN    $R3,<CHK2         BRANCH IF RECEIVE CHAN
002450          0A9A  E870 0405           LDR     $R6,=X'405'       SHOULD BE FOR TRANSMIT SIDE
002451          0A9C  8280 09ED           LD      <CCBSAV,=Z'1000'
                0A9E  1000
002452          0A9F  0586                BBF     >CHK3
002453          0AA0  E870 04A2           LDR     $R6,=X'04A2'
002454          0AA2  0F83                B       >CHK3
002455          0AA3  E870 0205   CHK2    LDR     $R6,=X'205'       P VALUE FOR GOOD COMPLETION
002456          0AA5  D956        CHK3    CMR     $R5,=$R6          COMPARE IS VS SB
002457          0AA6  0903                BE      >$+2+$AF
002458          0AA7  F380 125C           LNJ     $B7,<ERROR        INCORRECT OPERATION OF CHAN PROGRAM
```

```
002459  0AA9  8385                    JMP     $B5
002460                          *
002461                          * LCT TABLE OF XMIT TEST
002462                          *
002463  0AAA  0138              LCT1A   DC      =X'0138'            BYTE 56 = 1 FOR GNB, =0 FOR EOR
002464  0AAB  013F              LCT1    DC      =X'013F'            55 = COUNTER FOR BYTES
002465  0AAC  0537                      DC      X'0537'
002466  0AAD  0027                      DC      =X'27'              BYTE 39
002467  0AAE  0426                      DC      =X'0426'            BYTE 38
002468  0AAF  0125                      DC      X'0125'             PAUSE DISABLE
002469  0AB0  0000                      DC      0                   END OF TABLE
002470                          * LCT TABLE OF RCV TEST
002471  0AB1  0138              LCT2A   DC      =X'0138'            56 = FLG, 1 = GNB, = 0 FOR EOR
002472  0AB2  013F              LCT2    DC      =X'013F'            CONTER FOR BYTES
002473  0AB3  0537                      DC      X'0537'
002474  0AB4  0007                      DC      =X'07'              P VALUE START = 2
002475  0AB5  0206                      DC      =X'0206'
002476  0AB6  0105                      DC      X'0105'             PAUSE DISABLE
002477  0AB7  0000                      DC      0                   END OF LCT
002478                          *
002479                          * FILL SEND BUFFER WITH ASCENDING DATA + RECEIVE  BUFF WITH DEFAULT
002480                          * ONLY 10 LOC OF RECEIVE ARE FILLED WITH DEFAULT (5555).
002481                          *
002482                          * LNJ $B4,<FDATA
002483                          * DC DATA STARTING DATA
002484                          * DC RNG RANGE IN BYTES
002485                          * DC X    0 FOR LEFT BYTE START, 1 FOR RIGHT
002486                          *
002487  0AB8  8F40 0A5E         FDTA    SAVE    SAV3,=X'7440'       B1,R1,2,3,4
        0ABA  7440
002488  0ABB  9B80 1EBB                 LAB     $B1,<SDB            ADDRESS OF SEND DATA
002489  0ABD  C870 5555                 LDR     $R4,=X'5555'        DEFAULT FOR FIRST AND LAST
002490  0ABF  CF00 1EBB                 STR     $R4,<SDB
002491  0AC1  A874                      LDR     $R2,+$B4            STARTING DATA
002492  0AC2  B874                      LDR     $R3,+$B4            RANGE
002493  0AC3  8253                      NEG     =$R3
002494  0AC4  9874                      LDR     $R1,+$B4
002495  0AC5  A7DD              FDTALP  STH     $R2,$B1,+$R1        STORE BYTE
002496  0AC6  8AD2                      INC     =$R2
002497  0AC7  3780 0AC5                 BINC    $R3,<FDTALP
002498  0AC9  C791                      STH     $R4,$B1,$R1         DEFAULT AT END
002499  0ACA  1CF6                      LDV     $R1,=-10
002500  0ACB  9B80 2BAA                 LAB     $B1,<RTB            ADDRESS OF RECEIVE
002501  0ACD  A870 5555                 LDR     $R2,=Z'5555'        DEFAULT VALUE
002502  0ACF  AF71              FDTA1   STR     $R2,+$B1            STORE IN BUFFER
002503  0AD0  1760 0ACF                 BINC    $R1,<FDTA1
002504  0AD2  8FC0 0A44                 RSTR    SAV3,=X'7440'
        0AD4  7440
002505  0AD5  8384                      JMP     $B4
002506                          * ------------------------------------------------------------------
002507                          *
002508                          *
002509                          *       DATA INPUT AND CRC TEST
002510                          *
002511                          *-------------------------------------------------------------------
002512                          *
002513  0AD6  8870 4E20         CRCT    LDR     $R3,=A'N '
002514  0AD8  8F00 1581                 STR     $R3,<ERMG+1
002515                                  CALL    ZV$I,ZV$TC,TSTN
        0ADA  FBC0 0003
        0ADC  D380 0000         X
        0ADE  0F80
        0ADF  161D
002516  0AE0  C380 10FC                 LNJ     $B4,<GENITZ         MLCC GENERAL INITIALIZE
002517                          *
002518  0AE2  8C39                      LDV     $R6,=57             SET LOOP COUNT
002519  0AE3  EF00 14C8                 STR     $R6,<COUNT
002520  0AE5  9B80 1EBB                 LAB     $B5,<SDB            ADDRESS OF START OF BUFFER
002521  0AE7  DF80 14C5                 STB     $B5,<ADDS
002522  0AE9  A870 0100                 LDR     $R2,=X'100'
002523  0AEB  AF00 14D6                 STR     $R2,<RANGE1
002524  0AED  8753              NXCHC4  CL      =$R3
002525                          *
002526  0AEE  C380 10EC         NXCHC   LNJ     $B4,<FLN            FIND LINE ON
002527  0AF0  0FFD                      B       >NXCHC4             NO MORE, GO BACK TO FIRST
002528  0AF1  3B03                      BEVN    $R3,>NXCHC1
002529  0AF2  8AD3                      INC     =$R3
002530  0AF3  0FFB                      B       >NXCHC
002531  0AF4  9800 14C8         NXCHC1  LDR     $R1,<COUNT          GET LOOP COUNT
002532  0AF6  88D1                      DEC     =$R1
002533  0AF7  9F00 14C8                 STR     $R1,<COUNT
002534  0AF9  89D1                      CMZ     =$R1
002535  0AFA  0302                      BG      >NXCHC2
002536  0AFB  8382                      JMP     $B2                 END OF THIS TEST
002537  0AFC  8700 14E1         NXCHC2  CL      <MASK
002538  0AFE  8751                      CL      =$R1
002539  0AFF  8AD3                      INC     =$R3
002540                          *
002541                          *      PREPARE AND LOAD RAM INSTRUCTIONS
002542                          *
002543  0B00  9380 1069                 LNJ     $B1,<SDATA
002544  0B02  142A                      DC      <RCRT1
002545  0B03  006A                      DC      (ROD1-RCRT1)*2
002546  0B04  0200                      DC      X'200'              RAM ADDRESS
002547  0B05  0000                      DC      0                   EVEN BYTE START
002548                          *
002549                          *
002550  0B06  88D3                      DEC     =$R3                GET RECEIVE CHANNEL
002551  0B07  EB80 14A4                 LAB     $B6,<CRC16
002552  0B09  9B80 1EBB                 LAB     $B3,<SDB            ADDRESS OF CRC'S GENERATED
002553  0B0B  5C05                      LDV     $R5,=5              CHARACTER
002554  0B0C  E870 C002                 LDR     $R6,=Z'C002'        SET CONF. IN CPT FOR CRC16
002555  0B0E  EF00 0B37                 STR     $R6,<CPT3A
002556  0B10  6C08                      LDV     $R6,=8              NUMBER OF BITS IN CHARACTER
002557  0B11  F870 0100                 LDR     $R7,=X'100'          NUMBER OF CRC'S IN TABLE
002558  0B13  9380 0B3B                 LNJ     $B1,<TESC           GO DO TEST FOR CRC16
002559                          *
002560                          *
002561  0B15  5C01                      LDV     $R5,=1
002562  0B16  EB80 14A6                 LAB     $B6,<LRC8
002563  0B18  E870 00FF                 LDR     $R6,=X'FF'          CHANGE MASK
002564  0B1A  EF00 14E1                 STR     $R6,<MASK
002565  0B1C  E870 C602                 LDR     $R6,=Z'C602'        SET CONF. IN CPT FOR LRC8
```

```
002566  0B1E  EF00 0B37              STR   $R6,<CPT3A
002567  0B20  6C08                   LDV   $R6,=8
002568  0B21  9380 0B3B              LNJ   $B1,<TESC          GO DO TEST FOR LRC8
002569                          *
002570  0B23  0F81 000D              B     CRZR               SKIP CRC12
002571  0B25  5C05                   LDV   $R5,=5
002572  0B26  8700 14E1              CL    <MASK              RESET MASK
002573  0B28  EB80 14A7              LAB   $B6,<CRC12
002574  0B2A  E870 4402              LDR   $R6,=Z'4402'       SET CONF. IN CPT FOR CRC12
002575  0B2C  EF00 0B37              STR   $R6,<CPT3A
002576  0B2E  6C06                   LDV   $R6,=6             NUMBER OF BITS IN CHARACTER
002577  0B2F  9380 0B3B              LNJ   $B1,<TESC          DO TEST FOR CRC12
002578                          *
002579  0B31  3E02              CRZR  ADV   $R3,=2             INCREMENT TO NEXT INPUT CHANNEL
002580  0B32  0F80 0AEE              B     <NXCHC
002581                          *
002582                          * LCT FOR ABOVE TEST
002583                          *
002584  0B34  0206              CPT3  DC    Z'0206'          P, MSB, RCV
002585  0B35  0226                    DC    Z'0226'          P, XMIT , MSB
002586  0B36  0003                    DC    Z'0003'          PAUSE DISABLE, RCV
002587  0B37  C002              CPT3A DC    Z'C002'          CRC TYPE, RCV
002588  0B38  C022                    DC    Z'C022'          CRC, XMIT
002589  0B39  0004                    DC    Z'0004'          PAUSE DISABLE, XMIT
002590  0B3A  0000                    DC    0                END OF LCT
002591                          *
002592                          *
002593                          *
002594  0B3B  C380 1113         TESC  LNJ   $B4,<CRC           DO SIMULATED CRC
002595                          *
002596  0B3D  A870 0100              LDR   $R2,=X'100'
002597  0B3F  DBA8 14C5         TESCA LAB   $B5,*<ADDS.$R2     FORM ADDRESS FOR RECEIVE DATA
002598  0B41  DF80 14C5              STB   $B5,<ADDS
002599  0B43  9870 FF00              LDR   $R1,=-X'100'
002600  0B45  BB98 0000         X    LAB   $B3,*<ZV$HR.$R1
002601  0B47  0F00 0B3B              NOP   <TESC
002602  0B49  DDD3                   CMB   $B5,=$B3
002603  0B4A  0386                   BLE   >TESCC
002604  0B4B  BB80 1EBB              LAB   $B3,<SDB           RESET ADDRESS
002605  0B4D  BF80 14C5              STB   $B3,<ADDS
002606  0B4F  0FF0                   B     >TESCA             RESET MEM. POINTER, WE'RE OVER 64K.
002607  0B50  DF80 0B6D         TESCC STB   $B5,<CP1
002608  0B52  DF80 0B9E              STB   $B5,<COMP+4+2*$AF
002609  0B54  DF80 0B64              STB   $B5,<CP2+4+$AF
002610  0B56  A870 00F0              LDR   $R2,=X'F0'
002611  0B58  DBA8 14C5              LAB   $B5,*<ADDS.$R2     ADDRESS FOR LAST CCB
002612  0B5A  DF80 0B94              STB   $B5,<CP3
002613  0B5C  B380 0FCC              LNJ   $B3,<SETLCT        SEND LCT OUT FOR THIS LINE
002614  0B5E  0B34                   DC    <CPT3
002615                          *
002616                          CP2   CALL  ZV$F,$,DFLT,RANGE1
          0B5F  FBC0 0003
          0B61  D360 0000       X
          0B63  0F80
          0B64  0B5F
          0B65  14BE
          0B66  14D6
002617                          *
002618                          * IN THIS TEST 4 CCB'S ARE INITIALLY SET UP AND ONE ADDITIONAL IS
002619                          * SET UP AFTER DATA TRANSFERS ARE STARTED. THE BASE OF THE CCB'S
002620                          * IS MOVED UP WITH EACH NEW BLOCK XFER SO THAT ALL AVAILABLE
002621                          * MEMORY WILL HAVE DATA TRANSFERED TO IT. THE CCB'S ARE AS FOLLOWS:
002622                          *
002623                          *        CCB  CPU ADD   RANGE (WORDS)
002624                          *
002625                          *         1    B          80
002626                          *         2    B+80       40
002627                          *         3    B+C0       20
002628                          *         4    B+E0       10
002629                          *         5    B+F0       10
002630                          *
002631                          * IN THE ABOVE B = @ADDS@. B STARTS AT 1900(HEX) AND
002632                          * INCREASES BY 100(HEX) AFTER EACH SET OF 5 CCB'S.
002633                          *
002634  0B67  7CFC                   LDV   $R7,=-4
002635  0B68  C870 0100              LDR   $R4,=X'100'
002636  0B6A  8751              TESB  CL    =$R1
002637  0B6B  C380 1189              LNJ   $B4,<MCCB          FORM CCB FOR CHANNEL RUN
002638  0B6D  14C7              CP1   DC    <DUMMY           CPU DATA ADDRESS
002639  0B6E  0040                    DC    X'0040'          SET FOR NO RUPT
002640                          *
002641  0B6F  4041                    SUR   $R4,1            SHIFT OVER RANGE
002642  0B70  9854                    LDR   $R1,=$R4
002643  0B71  BB98 0B6D              LAB   $B3,*<CP1.$R1
002644  0B73  BF80 0B6D              STB   $B3,<CP1           SET FOR NEXT CCP
002645  0B75  7780 0B6A              BINC  $R7,<TESB
002646                          *
002647  0B77  7CFB                   LDV   $R7,=-5
002648  0B78  8756                   CL    =$R6
002649                          *
002650  0B79  B380 0628         TESF  LNJ   $B3,<STRTIO        START I/O
002651                          *
002652                          *
002653  0B7B  C380 1161         TESG  LNJ   $B4,<INXT          INPUT NEXT STATUS
002654  0B7D  0F88                    B     >TESE
002655  0B7E  C800 1561         TESH  LDR   $R4,<CONT6
002656  0B80  C380 1173               LNJ   $B4,<CGSCH
002657  0B82  8755                   CL    =$R5
002658  0B83  8055                   IO    =$R5,=$R4        INPUT STATUS
          0B84  0054
002659                          *
002660                          *
002661  0B85  82D5              TESE  LB    =$R5,=Z'1000'    SEE IF STATUS COMPLETE
          0B86  1000
002662  0B87  0506                   BBT   >TESD
002663  0B88  0F00 0B79              NOP   <TESF
002664  0B8A  67F4                   BINC  $R6,>TESH        TRY AGAIN
002665                          *
002666  0B8B  F380 125C              LNJ   $B7,<ERROR         CCB NOT COMPLETE AFTER CHANNEL PROG. EXECUTIO
002667  0B8D  7DFB              TESD  CMV   $R7,=-5
002668  0B8E  0988                   BNE   >TESJ
002669  0B8F  C870 0020              LDR   $R4,=X'20'         RANGE
002670  0B91  8751                   CL    =$R1               CLEAR INDEX
```

```
002671  0B92  C380 1189            LNJ   $B4,<MCCB            MAKE CCB
002672  0B94  14C7          CP3   DC    <DUMMY               FILLED IN BY PROGRAM
002673  0B95  0060                  DC    X'0060'              LAST BLOCK, VALID
002674  0B96  7780 0B7B      TESJ  BINC  $R7,<TESG            TEST FOR LAST
002675                        *
002676                        COMP  CALL  ZV$C,SDB,$,MASK,RANGE1,ERAR
        0B98  FBC0 0003
        0B9A  D380 0000   X
        0B9C  0F80
        0B9D  1EBB
        0B9E  0B98
        0B9F  14E1
        0BA0  14D6
        0BA1  14E3
002677  0BA2  89C0 0940            CMZ   ERAR
002678  0BA4  090F                  BE    >NOERC               IF ERROR HALT WITH:
002679  0BA5  DC80 0B9E            LDB   $B5,<COMP+4+2*$AF
002680  0BA7  D800 14E5            LDR   $R5,<ERAR+2          '15' IN $R5
002681  0BA9  E800 14E4            LDR   $R6,<ERAR+1          'SHOULD BE' IN $R6
002682  0BAB  F800 14E3            LDR   $R7,<ERAR            WITH THE LOCATION OF ERROR IN BLOCK
002683  0BAD  F380 125C            LNJ   $B7,<ERROR           ERROR IN CRC GENERATION, $B6 POINTS TO TYPE
002684                        * OF CRC+1. $B5 POINTS TO START OF RECEIVE BUFFER
002685                              CALL  ZV$C0                CONTINUE COMPARISON
        0BAF  FBF0 0001
        0BB1  D380 0000   X
002686  0BB3  C380 1177      NOERC  LNJ   $B4,<CHCT           CHANNEL CONTROL
002687  0BB5  0F82                  B     >$+2
002688  0BB6  0000                  DC    0                  CHANNEL INIT.
002689                        *
002690  0BB7  F870 0100            LDR   $R7,=X'100'
002691  0BB9  BB60 1EBB            LAB   $B3,<SDB
002692  0BBB  8381                  JMP   $B1                JUMP TO DO NEXT CRC TYPE
002693                        *-------------------------------------------------------------
002694                        *
002695                        *
002696                        *
002697                        *          DATA OUTPUT TEST
002698                        *
002699                        *
002700                        *-------------------------------------------------------------
002701  0BBC  B870 5020      OTPT  LDR   $R3,=A'P '
002702  0BBE  BF00 1581            STR   $R3,<ERMG+1
002703                              CALL  ZV$T,ZV$TC,TSTP
        0BC0  FBC0 0003
        0BC2  D380 0000   X
        0BC4  0F80
        0BC5  162A
002704  0BC6  8753                  CL    =$R3               INITIALIZE CHANNEL
002705  0BC7  8751                  CL    =$R1
002706  0BC8  C380 10FC            LNJ   $B4,<GENITZ        GENERAL INITIALIZE
002707                        *
002708  0BCA  C380 10EC      OTPT2 LNJ   $B4,<FLN           FIND LINE ON
002709  0BCC  8382                  JMP   $B2                NO SUBCH ON NO TEST
002710  0BCD  3BB3                  BODD  $R3,>OTPT4         GET WRITE SUBCH. ONLY
002711  0BCE  8AD3                  INC   =$R3
002712  0BCF  0FFB                  B     >OTPT2
002713  0BD0  8700 14E1      OTPT4 CL    <MASK
002714                        *
002715                        *    LOAD AND PACK RAM INSTRUCTIONS
002716                        *
002717  0BD2  9380 1069            LNJ   $B1,<SDATA
002718  0BD4  145F                  DC    <ROD1
002719  0BD5  003C                  DC    (VALTST-ROD1)*2      NUMBER OF RAM INSTRUCTIONS
002720  0BD6  0200                  DC    X'200'               RAM STARTING ADDRESS
002721  0BD7  0000                  DC    0                    EVEN START ADDRESS
002722                        *
002723                        *
002724                        *
002725                        *          FORM RANDOM DATA TABLE TO BE OUTPUT
002726                        *
002727  0BD8  BB80 1EBB            LAB   $B3,<SDB             LOAD $B3 WITH ADDRESS OF SDB
002728  0BDA  EB80 14A4            LAB   $B6,<CRC16           LOAD $B6 WITH ADDRESS OF CRC16
002729  0BDC  5C0A                LDV   $R5,10               INITIAL CHARACTER
002730  0BDD  6C0B                LDV   $R6,=11              NUMBER OF BITS IN CHAR.
002731  0BDE  F870 01F4            LDR   $R7,=500             THE NUMBER OF WORDS MADE IN TABLE
002732  0BE0  C380 1113            LNJ   $B4,<CRC
002733                        *
002734  0BE2  B380 0FCC      OTPT5 LNJ   $B3,<SETLCT          SEND LCT BYTES OUT
002735  0BE4  0B34                  DC    <CPT3
002736                        *
002737  0BE5  C870 01F4            LDR   $R4,=500             RANGE OF BYTES TO BE LOADED FOR CRC
002738  0BE7  C380 1189            LNJ   $B4,<MCCB            FORM CCB FOR CHANNEL RUN
002739  0BE9  1EBB                  DC    <SDB                 XMIT BUFFER ADDRESS
002740  0BEA  0040                  DC    X'0040'              NO RUPI
002741                        *
002742  0BEB  C870 00C8            LDR   $R4,=200
002743  0BED  C380 1189            LNJ   $B4,<MCCB            MAKE CCB
002744  0BEF  1FB5                  DC    <SDB+250
002745  0BF0  0040                  DC    X'40'
002746                        *
002747  0BF1  C870 0064            LDR   $R4,=100
002748  0BF3  C380 1189            LNJ   $B4,<MCCB            MAKE CCB
002749  0BF5  2019                  DC    <SDB+350             CPU ADDRESS
002750  0BF6  0040                  DC    X'40'                VALID BIT ONLY
002751                        *
002752  0BF7  C870 0064            LDR   $R4,=100
002753  0BF9  C380 1189            LNJ   $B4,<MCCB            MAKE CCB
002754  0BFB  204B                  DC    <SDB+400             CPU ADDRESS
002755  0BFC  0040                  DC    X'40'                VALID
002756                        *
002757                        *
002758  0BFD  B380 0628            LNJ   $B3,<STRTIO          START I/O
002759                        *
002760                        * WAIT FOR FIRST CCB TO GET READY AND THEN GIVE 5TH CCB
002761                        *
002762  0BFF  C380 11C8            LNJ   $B4,<TEST            WAIT FOR STATUS COMPLETE
002763  0C01  0F82                  B     >$+2                 RETURN
002764  0C02  003C                  DC    60                   1/2 SEC TIMEOUT
002765                        *
002766  0C03  8205                  LB    =$R5,=Z'1000'        GET STAT COMPLETE BIT
        0C04  1000
002767  0C05  0500 0C09            BBT   <OTPT7               IF TRUE GO SET UP CCB
002768  0C07  F380 125C            LNJ   $B7,<ERROR           NO CCB COMPLETE ON TRANSMIT
```

```
002769                                  *
002770  0C09  C870 0064                OTPT7   LDR     $R4,=100                RANGE
002771  0C0B  C380 1189                        LNJ     $B4,<MCCB               MAKE CCB
002772  0C0D  207D                             DC      <SDB+450                CPU ADDRESS
002773  0C0E  0060                             DC      X'60'                   VALID,LAST BLOCK
002774                                  *
002775  0C0F  C380 1205                        LNJ     $B4,<DLAY               TIME DELAY
002776                                  *
002777  0C11  E853                             LDR     $R6,=$R3                CALCULATE LCT CRC RESIDUE ADDRESS
002778  0C12  6F20                             MLV     $R6,=X'20'
002779  0C13  6E03                             ADV     $R6,=3
002780  0C14  EF40 0010                        STR     $R6,OTPT3
002781                                  *
002782                                  *       ACCUMULATE CRC OF RANDOM TABLE FORMED EARLIER
002783                                  *
002784  0C16  5CFF                             LDV     $R5,=-1
002785  0C17  BB80 1EBB                        LAB     $B3,<SDB
002786  0C19  EB80 14A4                        LAB     $B6,<CRC16
002787  0C1B  6C08                             LDV     $R6,8                   8 BIT CHARACTER
002788  0C1C  F870 01F4                        LDR     $R7,=500                NUMBER IN TABLE
002789  0C1E  C380 1113                        LNJ     $B4,<CRC
002790                                  *
002791  0C20  88D3                             DEC     =$R3                    GET READ CHANNEL
002792  0C21  9380 1098                        LNJ     $B1,<RDATA              DO BLOCK READ
002793  0C23  14C7                             DC      <DUMMY                  TO HERE
002794  0C24  0002                             DC      2                       RANGE =2
002795  0C25  0000                      OTPT3  DC      0                       RAM ADDRESS GOS HERE
002796  0C26  0000                             DC      0                       EVEN BYTE BOUNDRY
002797                                  *
002798                                  *
002799  0C27  D800 14C7                        LDR     $R5,<DUMMY
002800  0C29  5018                             SCL     $R5,8
002801  0C2A  D900 14D3                        CMR     $R5,<CRCAC              ARE CRC'S EQUAL
002802  0C2C  0905                             BE      >PRFR1                  YES DO NEXT CHANNEL
002803  0C2D  E800 14D3                        LDR     $R6,<CRCAC              'IS' IN $R5 'SB' IN $R6
002804                                  *
002805  0C2F  F380 125C                        LNJ     $B7,<ERROR              OUTPUT ERROR
002806                                  *
002807  0C31  3E03                      PRFR1  ADV     $R3,=3                  INCREMENT TO NEXT CHANNEL
002808  0C32  C380 10EC                 OTPT1  LNJ     $B4,<FLN
002809  0C34  8382                             JMP     $B2
002810  0C35  3B80 0BE2                        BODD    $R3,<OTPT5
002811  0C37  8AD3                             INC     =$R3
002812  0C38  0FFA                             B       >OTPT1
002813                                  *----------------------------------------------------------------
002814                                  *
002815                                  *
002816                                  * INTERRUPT DESCRIPTION
002817                                  *
002818                                  ****
002819                                  * BEFORE RUNNING UNDER INTERRUPT MODE THE FOLLOWING
002820                                  * TABLES MUST BE SET UP. ALL TABLE CONTAIN 16 ENTITIES
002821                                  * CORRESPONDING TO THE 16 CHANNELS IN SEQUENCE. AN ENTITY
002822                                  * IS EITHER A DATA OR ADDRESS CONSTANT.
002823                                  *
002824                                  *
002825                                  *       INMB - SET UP TO CONTAIN THE NEG. NUMBER OF
002826                                  *              INTERRUPTS EXPECTED PER CHANNEL.
002827                                  *              IF IT IS SET TO A POS. NUMBER IT WILL
002828                                  *              BE ASSUMED THAT NO INTERRUPTS ARE EXPECTED
002829                                  *              ON THAT CHANNEL.
002830                                  *
002831                                  *       ILV - SET UP TO CONTAIN THE LEVEL ASSIGNED
002832                                  *             TO EACH CHANNEL.
002833                                  *
002834                                  *       IST - SET UP TO CONTAIN THE STATUS EXPECTED
002835                                  *             ON EACH CHANNEL AFTER INTERRUPTS.
002836                                  *
002837                                  *
002838                                  * AFTER AN INTERRUPT TEST HAS BEEN TERMINATED (NORMALLY OR OTHERWISE)
002839                                  * THERE IS ONE OTHER TABLE CONTAINING USEFUL INFORMATION
002840                                  *
002841                                  *       PRIST  - CONTAINS ORDER IN WHICH FIRST 16
002842                                  *              INTERRUPTS OCCURED.
002843                                  *
002844                                  * THE PROGRAM RUNS AT 5 LOGICALLY DISTINCT LEVELS:
002845                                  *
002846                                  *       LEVEL 0 - USED FOR SETUPS AND DATA CHECKING.
002847                                  *             IT IS EXITED BY SCHEDULING LEVEL 63
002848                                  *             BY MEANS OF A "LEV" INSTRUCTION
002849                                  *
002850                                  *       LEVEL 3 - REAL TIME CLOCK TIMEOUT. MAJOR TESTS
002851                                  *             WITH INTERRUPTS TERMINATE BY A RTC
002852                                  *             TIMEOUT. THIS LEVEL EXITS BY SCHEDULING
002853                                  *             LEVEL 0.
002854                                  *
002855                                  *
002856                                  *       LEVEL 4 -  CCB DISPATCHER AND INTERRUPT MONITOR.
002857                                  *             THIS ROUTINE DOES STATUS CHECKING AND
002858                                  *             DISPATCHES A NEW CCB FOR A CHANNEL
002859                                  *             IF NEEDED. LEVEL 4 EXITS TO LEV 63 BY
002860                                  *             A "LEV" INSTRUCTION.
002861                                  *
002862                                  *       LEVELS 5-62   -  16 OF THESE LEVELS (VARIABLE) ARE USED
002863                                  *             FOR THE 16 CHANNELS. EACH LEVEL
002864                                  *             POINTS TO THE SAME RE-ENTRANT
002865                                  *             CHANNEL INTERRUPT HANDLER.
002866                                  *
002867                                  *       LEVEL 63 -  A "DO NOTHING-WAIT" ROUTINE
002868                                  *             RESIDES AT THIS LEVEL.
002869                                  *
002870                                  * TO USE THE PROGRAM INTERRUPT STRUCTURE THE FOLLOWING
002871                                  * STEPS MUST BE TAKEN
002872                                  *
002873                                  *       1. USE ROUTINE INTITZ. THIS INITIALIZES ALL
002874                                  *       TABLES.
002875                                  *
002876                                  *       2. FILL TABLES WITH INFORMATION.
002877                                  *
002878                                  *       3. SETUP INTERRUPT VECTORS. (INITITZ PUTS
002879                                  *       VECTORS FOR ERROR HANDLER IN LEVELS 5-62. IT
002880                                  *       SETS UP VECTORS FOR LEV 0,3,4, AND 5 TO THIER
002881                                  *       CORRECT VALUE.)
```

```
002882                          *       4. USE ROUTINE "RTC" TO SET UP RTC TO
002883                          *            DESIRED TIMEOUT VALUE AND TO START RTC.
002884                          *
002885                          *       5. TURN ON DESIRED CHANNELS.
002886                          *
002887                          *       6. SCHEDULE AND DISPATCH LEVEL 63.
002888                          *
002889                          * WHEN THE RTC TIMEOUT IS COMPLETE THE PROGRAM
002890                          * WILL RESUME AT THE INSTRUCTION FOLLOWING THE "LEV"
002891                          * WHICH DISPATCHED LEVEL 63.
002892                          *
002893                          *
002894                          * *****************
002895                          *
002896                          * FIRST INTERRUPT TEST
002897                          *
002898  0C39  B870 5220         INST1   LDR     $R3,=A'R '
002899  0C3B  BF00 1581                 STR     $R3,<ERMG+1
002900                                  CALL    ZV$T.ZV$TC,TSTR
        0C3D  FBC0 0003
        0C3F  D380 0000 X
        0C41  0F80
        0C42  1633
002901  0C43  8753                      CL      =$R3
002902  0C44  8E70 0080                 LEV     =Z'0080'                SWITCH TO LEVEL 0
002903  0C46  C380 10FC         INLOOP  LNJ     $B4,<GENITZ             GENERAL INITIALIZE TO MLCC
002904  0C48  C380 10EC                 LNJ     $B4,<FLN                FIND ACTIVE LINE
002905  0C4A  8382                      JMP     $B2                     NO MORE LEFT
002906  0C4B  9380 0F18                 LNJ     $B1,<INTITZ             INITIALIZE INTERRUPTS
002907  0C4D  C380 0E18                 LNJ     $B4,<CHINTZ             INITIALIZE CHANNEL
002908  0C4F  C380 1205                 LNJ     $B4,<DLAY               DELAY 25MS
002909                          *
002910  0C51  B380 0FCC                 LNJ     $B3,<SETLCT             OUTPUT LCT
002911  0C53  0D15                      DC      <INITLCT                FROM HERE
002912                          *
002913                          * SET UP EXPECTED STATUS TABLE
002914  0C54  9870 5020                 LDR     $R1,=Z'5020'            STATUS COMPLETE
002915  0C56  9F30 2690                 STR     $R1,<IST.$R3
002916                          *
002917  0C58  1C06                      LDV     $R1,=6                  PUT IN LEVEL OF 6 FOR CHANNEL
002918  0C59  9F30 26A0                 STR     $R1,<ILV.$R3
002919                          * SEND OUT CHANNEL PROGRAM
002920  0C5B  BF00 14CD                 STR     $R3,<TPR
002921  0C5D  8953                      LBT     =$R3,=X'1'
        0C5E  0001
002922  0C5F  9380 1069                 LNJ     $B1,<SDATA              SEND PROGRAM
002923  0C61  0CDC                      DC      <GNDPRG
002924  0C62  0024                      DC      36                      20 BYTE
002925  0C63  0200                      DC      =X'200'                 RAM ADDRESS
002926  0C64  0000                      DC      0                       STARTS ON EVEN BYTE IN CPU
002927  0C65  B800 14CD                 LDR     $R3,<TPR
002928                          * SEND OUT CCB. IN THIS FIRST TEST THE "1" BIT (BIT 0) IS
002929                          * SET TO A "0". NO INTERRUPT SHOULD OCCUR FOR THIS FIRST TEST
002930  0C67  C870 0004                 LDR     $R4,=4                  ANY RANGE
002931  0C69  8751                      CL      =$R1
002932                          *
002933  0C6A  C380 1189                 LNJ     $B4,<MCCB               SEND CCB
002934  0C6C  2BAA                      DC      <RTD                    ANY ADDRESS
002935  0C6D  0060                      DC      =X'0060'                LAST BLOCK BIT ONLY
002936                          *
002937                          * SEND OUT INTERRUPT LEVEL 6 + RETURN CHANNEL NUMBER TO LCT
002938                          *
002939  0C6E  C380 0DF6                 LNJ     $B4,<LEVRCH             OUTPUT LEVEL + RETURN CHANNEL
002940                          *
002941                          * SET UP INTERRUPT VECTOR IN CP
002942                          *
002943  0C70  C380 10DB                 LNJ     $B4,<SETVEC             SETS UP VECTOR FOR LEVEL
002944                          *                                       SET UP IN TABLE ILV
002945  0C72  C800 155D                 LDR     $R4,<CONT2              GET INPUT RANGE FC
002946  0C74  C380 1173                 LNJ     $B4,<CGSCH              PUT IN ADDRESS
002947  0C76  CF00 0C86                 STR     $R4,<RNFC
002948                          *
002949                          * TURN ON CHANNEL
002950                          *
002951  0C78  9B80 0C87                 LAB     $B1,<LV63TA             GET ADDRESS TO CONTINUE
002952  0C7A  9F80 0EEC                 STB     $B1,<L63SV2             PUT IN LEVEL 63 SAVE AREA
002953  0C7C  C380 124C                 LNJ     $B4,<RTC                SETUP RTC TO INTERRUPT
002954  0C7E  000A                      DC      X'000A'                  ************ X'000A'FTER 50 MSEC
002955  0C7F  8F00 0EEE                 SAVE    <L63SV3,=X'7884'
        0C81  7884
002956  0C82  8E70 803F                 LEV     =Z'803F'                SUSPEND CURRENT AND GO TO 63
002957  0C84  0F80 0C92                 B       <BKHR
002958  0C86  0000              RNFC    DC      0                       RANGE CONTROL WORD
002959                          *
002960  0C87  B380 10B0         LV63TA  LNJ     $B3,<TRNON              TURN ON CHANNEL
002961  0C89  4000                      DC      =X'4000'                CHANNEL CONTROL WORD - START IO
002962  0C8A  0000                      DC      0                       WORKS FOR XMIT + RECEIVE
002963  0C8B  8E70 00BF         IDLP    LEV     =Z'00BF'                DUMM LEV TO GENERATE RINT
002964  0C8D  8055                      IO      =$R5,<RNFC              INPUT RANGE
        0C8E  0000 0C86
002965  0C90  0F80 0C8B                 B       <IDLP
002966                          *
002967                          * PROGRAM RETURNS TO LEVEL 0 HERE IF NO ERRORS
002968  0C92  0005              BKHR    RTCF                            TURN OFF REAL TIME CLOCK
002969                          *
002970                          * CHECK STATUS IS COMPLETE
002971                          *
002972  0C93  C380 1161                 LNJ     $B4,<INXT               INPUT NEXT STATUS TO R5
002973  0C95  82D5                      LB      =$R5,=Z'1000'
        0C96  1000
002974  0C97  0503                      BB1     >$+2+$AF
002975  0C98  F380 125C                 LNJ     $B7,<ERROR              STATUS NOT COMPLETE AFTER
002976                          *                                       "GNB" ISSUED
002977                          *
002978                          * SET UP FOR NEXT TEST. SET UP CCB WITH 1 BIT = 1 (PERMITTING
002979                          * INTERRUPTS)
002980                          *
002981  0C9A  8700 14CC                 CL      <FLAG                   FLAG CLEARED FOR FIRST TIME THROUGH
002982  0C9C  9380 0F18         INST2   LNJ     $B1,<INTITZ             INITIALIZE PROGRAM FOR INTERRUPTS
002983  0C9E  9870 9020                 LDR     $R1,=Z'9020'
002984  0CA0  8980 14CC                 CMZ     <FLAG
002985  0CA2  0983                      BNE     >INST4                  BRANCH IF SECOND TIME THRU
002986                          *
```

```
002987  0CA3  9870 5020            LDR     $R1,=Z'5020'          STATUS COMPLETE
002988  0CA5  0F81 0003    INST4   B       NEXT4                 ****************TEMP. CHANGE
002989  0CA7  9670 0020            XOR     $R1,=Z'0020'
002990  0CA9  9F30 2690    NEXT4   STR     $R1,<IST.$R3          SET EXPECTED STATUS
002991                      *
002992  0CAB  1C06                 LDV     $R1,=6                CHANNEL TO HAVE LEVEL 6
002993  0CAC  9F30 26A0            STR     $R1,<ILV.$R3
002994                      *
002995  0CAE  1CFF                 LDV     $R1,=-1
002996  0CAF  9F30 2680            STR     $R1,<INMB.$R3         SET TO EXPECT ONE INTERRUPT
002997  0CB1  C380 10DB            LNJ     $B4,<SETVEC           SET UP RUPT VECTOR IN CP
002998                      *
002999                      * SEND OUT CCB WITH 1 BIT SET
003000                      * THIS CCB NOT USED SECOND TIME THRU LOOP
003001                      *
003002  0CB3  4C04                 LDV     $R4,=4                ANY RANGE
003003  0CB4  8751                 CL      =$R1                  CLEAR BYTE INDEX
003004  0CB5  C380 1189            LNJ     $B4,<MCCB             MAKE CCB
003005  0CB7  2BAA                 DC      <RTB                  ANY ADDRESS
003006  0CB8  00E0                 DC      =Z'00E0'              INTERRUPT,<LAST BLOCK
003007  0CB9  9B80 0C87            LAB     $B1,<LV63TA           GET ADDRESS TO CONTINUE
003008  0CBB  9F80 0EEC            STB     $B1,<L63SV2           PUT IN LEVEL 63 SAVE AREA
003009  0CBD  C380 124C            LNJ     $B4,<RTC              SET UP RTC TO INTERRUPT
003010  0CBF  0006                 DC      6                     AFTER 50 MSEC
003011  0CC0  8F00 0EEE            SAVE    <L63SV3,=X'7884'
        0CC2  7884
003012  0CC3  8E70 803F            LEV     =Z'803F'              SUSPEND CURRENT AND GO TO 63
003013                      *
003014                      * RETURN HERE AFTER INTERRUPTS HAVE OCCURED
003015                      *
003016  0CC5  0005                 RTCF                          TURN OFF REAL TIME CLOCK
003017                      *                                   "GNB" ISSUED
003018                      * CHECK THAT INTERRUPT DID OCCUR
003019                      *
003020  0CC6  9380 0FB9            LNJ     $B1,<CHLV             CHECK 1 INTERRUPT OCCURED
003021  0CC8  9800 14CC            LDR     $R1,<FLAG
003022  0CCA  198F                 BNEZ    $R1,>INST3            BRANCH IF SECOND TIME THRU
003023  0CCB  8A80 14CC            INC     <FLAG                 SET FIRST TIME THRU FLAG
003024  0CCD  4C01                 LDV     $R4,=1                ANY RANGE
003025  0CCE  C380 1189            LNJ     $B4,<MCCB             MAKE CCB
003026  0CD0  2BAA                 DC      <RTB                  ANY ADDRESS
003027  0CD1  0060                 DC      X'60'                 LAST BLOCK, VALID
003028                      *
003029  0CD2  B380 0FCC            LNJ     $B3,<SETLCT           SEND OUT LCT BYTES
003030  0CD4  0CD7                 DC      <SFLG                 FROM HERE
003031  0CD5  0F80 0C9C            B       <INST2
003032                      *
003033  0CD7  0317          SFLG   DC      X'0317'                SETS LCT X'17' NON ZERO
003034  0CD8  0000                 DC      0                     END OF LIST
003035                      *                                   ON THIS CHANNEL.
003036  0CD9  8AD3          INST3  INC     =$R3
003037  0CDA  0F80 0C46            B       <INLOOP
003038                      *
003039                      * CHANNEL PROGRAM FOR ABOVE TEST
003040                      *
003041  0CDC  BD31          GNBPRG DC      =Z'BD31'
003042  0CDD  20BD                 DC      =Z'20BD'
003043  0CDE  3100                 DC      =Z'3100'
003044  0CDF  C605                 DC      =Z'C605'
003045  0CE0  0101                 DC      =Z'0101'
003046  0CE1  5A01                 DC      =Z'5A01'
003047  0CE2  5D26                 DC      =Z'5D26'
003048  0CE3  FBBD                 DC      =Z'FBBD'
003049  0CE4  3120                 DC      =Z'3120'
003050  0CE5  BD31                 DC      =Z'BD31'
003051  0CE6  00BD                 DC      =Z'00BD'
003052  0CE7  3180                 DC      =Z'3180'
003053  0CE8  01BD                 DC      =Z'01BD'
003054  0CE9  3120                 DC      =Z'3120'
003055  0CEA  BD31                 DC      =Z'BD31'
003056  0CEB  0001                 DC      =Z'0001'
003057  0CEC  BD31                 DC      =Z'BD31'
003058  0CED  0001                 DC      =Z'0001'
003059  0CEE  0000                 DC      0
003060                      *
003061                      *-----------------------------------------------------------------
003062                      * * * * * * * *
003063                      * DATA SET SCAN TEST
003064                      *
003065                      *-----------------------------------------------------------------
003066  0CEF  B870 5320     DSST   LDR     $R3,=A'S '
003067  0CF1  BF00 1581            STR     $R3,<ERMG+1
003068                             CALL    ZV$T,ZV$TC,TSTS
        0CF3  FBC0 0003
        0CF5  D380 0000     X
        0CF7  0F80
        0CF8  163E
003069  0CF9  8753                 CL      =$R3
003070  0CFA  C380 10EC     DSSTLP LNJ     $B4,<FLN              FIND ACTIVE LINE
003071  0CFC  8382                 JMP     $B2                   END OF TEST
003072  0CFD  C380 10FC            LNJ     $B4,<GENITZ           INITIALIZE
003073  0CFF  C380 1205            LNJ     $B4,<DLAY             DELAY 25 MS
003074                      * LOAD CHANNEL PROGRAM FOR DATA SET SCAN
003075  0D01  BF00 14CD            STR     $R3,<TPR
003076  0D03  8953                 LBT     =$R3,=X'1'            MAKE SURE WE HAVE A XMIT CHANNEL
        0D04  0001
003077  0D05  9380 1069            LNJ     $B1,<SDATA            SEND PROGRAM
003078  0D07  0DF1                 DC      <SCNPRG               POINTER TO SCAN PROGRAM
003079  0D08  000A                 DC      =Z'000A'              RANGE
003080  0D09  0200                 DC      =X'200'               RAM ADDRESS
003081  0D0A  0000                 DC      0                     START ON EVEN BYTE
003082                      *
003083  0D0B  B800 14CD            LDR     $R3,<TPR              GET BACK TEST CHANNEL
003084                      *
003085                      * OUTPUT LCT DATA
003086                      *
003087                      * DATA SCAN MASK = ALL ZERO'S (0)
003088                      * MAKE CLA STATUS = ALL ONES (F)
003089                      *
003090  0D0D  B380 0FCC            LNJ     $B3,<SETLCT           SET LCT
003091  0D0F  0D11                 DC      <SCTEST
003092  0D10  0F90                 B       >SCTST1
003093                      *
```

```
003094  0D11  000F                SCTEST DC     =Z'000F'          RECEIVE MASK
003095  0D12  002F                       DC     =Z'002F'          XMIT MASK
003096  0D13  FF0E                       DC     =Z'FF0E'          RECEIVE CLA STATUS SET TO FF
003097  0D14  FF2E                       DC     =Z'FF2E'          XMIT CLA STATUS SET TO FF
003098  0D15  0206                INTLCT DC     =Z'0206'          RCV P COUNTER,<MSB
003099  0D16  0007                       DC     =Z'0007'          RCV P COUNTER,<LSB
003100  0D17  0226                       DC     =Z'0226'          XMIT P COUNTER,<MSB
003101  0D18  0027                       DC     =Z'0027'          XMIT P COUNTER,<LSB
003102  0D19  0010                       DC     =Z'0010'          RCV STAT (LCT)
003103  0D1A  0011                       DC     =Z'0011'
003104  0D1B  0030                       DC     =Z'0030'          XMIT STATUS (LCT)
003105  0D1C  0031                       DC     =Z'0031'
003106  0D1D  0008                       DC     =Z'0008'          RCV SCAN CONTROL
003107  0D1E  0028                       DC     =Z'0028'          XMIT SCAN CONTROL
003108  0D1F  0000                       DC     0
003109                            * START SCAN AND CHECK THAT MASK OF 0 INHIBITS ITS OPERATION
003110  0D20  9870 C008           SCTST1 LDR    $R1,=Z'C008'      DATA SCAN = 1, SET LCT STAT = 1
003111  0D22  3B02                       BEVN   $R3,>$+2
003112  0D23  1E20                       ADV    $R1,=X'20'        MODIFY FOR TRANSMIT
003113  0D24  C380 0B0C                  LNJ    $B4,<OUTLCT       SEND OUT CHANNEL COMMAND BYTE
003114  0D26  C380 1215                  LNJ    $B4,<DLAYLG       DELAY 250MS
003115  0D28  C380 084E                  LNJ    $B4,<ILCTST       INPUT LCI STATUS TO R5
003116  0D2A  82D5                       LB     =$R5,=X'10'       GET DATA SET CHANGE BIT
        0D2B  0010
003117  0D2C  0583                       BBF    >$+2+$AF
003118  0D2D  F380 125C                  LNJ    $B7,<ERROR        DATA SET SCAN BIT SET WITH
003119                            *                               MASK OF 0
003120                            *
003121                            * READ CLA STATUS BY INPUT DATA SET STATUS
003122  0D2F  9870 000E                  LDR    $R1,=X'E'
003123  0D31  3B02                       BEVN   $R3,>$+2          BRANCH IF RECEIVE CHAN
003124  0D32  1E20                       ADV    $R1,=X'20'        FORM CLA ADDRESS
003125  0D33  9F57                       STR    $R1,=$R7
003126  0D34  C380 0EC6                  LNJ    $B4,<DSSTA        READ DATA SET STATUS
003127  0D36  D570 FF00                  AND    $R5,=Z'FF00'      STRIP OFF UNUSED BYTE
003128  0D38  D657                       XOR    $R5,=$R7          FORM STATUS WITH ADDRESS
003129  0D39  DF00 14CF                  STR    $R5,<TEMP1
003130  0D3B  8655                       CPL    =$R5              FORM COMPLEMENT OF STATUS
003131  0D3C  D570 FF00                  AND    $R5,=Z'FF00'      STRIP ADDRESS
003132  0D3E  D657                       XOR    $R5,=$R7          OR IN ADDRESS
003133                            * OUTPUT COMPLEMENT OF NOMINAL CLA STATUS. DATA SCAN IS STILL
003134                            * PROGRESSING.
003135  0D3F  DF00 14D0                  STR    $R5,<TEMPS1       FIRST STORE AWAY DATA
003136  0D41  C380 0E0C                  LNJ    $B4,<OUTLCT       OUTPUT BYTE TO LCT
003137  0D43  C3C0 04C1                  LNJ    $B4,DLAY          WAIT FOR FIRMWARE TO FINISH
003138                            *
003139                            * INPUT LCT STATUS AND CHECK DATA SET CHANGE IS RESET
003140                            *
003141  0D45  C380 084E                  LNJ    $B4,<ILCTST       INPUT LCI STATUS TO R5
003142  0D47  82D5                       LB     =$R5,=X'10'       GET DATA SET CHANGE BIT
        0D48  0010
003143  0D49  0583                       BBF    >$+2+$AF
003144  0D4A  F380 125C                  LNJ    $B7,<ERROR        DATA SET SCAN BIT SET WITH
003145                            *                               MASK OF 0
003146                            *
003147                            * START OF LOOP WHICH WILL TEST THE DATA SET MASK
003148                            *
003149  0D4C  9870 0400                  LDR    $R1,=X'400'
003150  0D4E  9F00 14E1                  STR    $R1,<MASK
003151                            *
003152                            *
003153                            *
003154  0D50  B3C0 027B           DSLOOP LNJ    $B3,SETLCT        SEND OUT LCT
003155  0D52  0D11                       DC     <SCTEST
003156  0D53  9800 14E1                  LDR    $R1,<MASK
003157  0D55  1001                       SOL    $R1,1
003158  0D56  0600 0D96                  BCT    <SCITIA           EXIT IF NO MORE BITS TO TEST
003159  0D58  9F00 14E1                  STR    $R1,<MASK         STORE MASK
003160  0D5A  9670 000F                  XOR    $R1,=X'F'         OR IN ADDRESS FOR RCV MASK
003161  0D5C  3B02                       BEVN   $R3,>$+2
003162  0D5D  1E20                       ADV    $R1,=X'20'        MODIFY IF XMIT
003163  0D5E  C380 0E0C                  LNJ    $B4,<OUTLCT       OUTPUT MASK (R1) TO LCT
003164                            * OUTPUT BYTE TO CLA POSITION IN LCT. THIS BYTE WILL TEST
003165                            * ONE BIT POSITION CORRESPONDING TO THE MASK WHICH HAS ONE BIT SET.
003166                            * EACH BIT POSITION IN THE STATUS MASK IS CHECKED ONCE
003167                            *
003168  0D60  C380 1205                  LNJ    $B4,<DLAY         DELAY 25 MSEC
003169  0D62  9800 14CF                  LDR    $R1,<TEMP1        GET CLA NORMAL STAT + ADDRESS
003170                            *                               WITH LCI ADDRESS
003171  0D64  9600 14E1                  XOR    $R1,<MASK         COMPLEMENT BIT TO TEST
003172  0D66  C380 0E0C                  LNJ    $B4,<OUTLCT       OUTPUT TO CLA STATUS
003173                            * SET UP FOR ERROR INTERRUPT
003174  0D68  C380 0DF6                  LNJ    $B4,<LEVRCH       SET LCI BYTES FOR LEVEL = 6 + RETURN CHAN
003175  0D6A  9380 0F18                  LNJ    $B1,<INTITZ       INITIALIZE INTERRUPTS
003176  0D6C  1C06                       LDV    $R1,=6
003177  0D6D  9F30 26A0                  STR    $R1,<ILV.$R3      SET FOR LEVEL 6 IN LEVEL TABLE
003178                            *
003179                            * SWITCH TO LEVEL 63
003180                            * DATA SET SCAN WILL TURN ON WITH CONTROL = "D0"
003181  0D6F  9B80 0D7C                  LAB    $B1,<LV63TB
003182  0D71  9F80 0EEC                  STB    $B1,<L63SV2       SET PVECTOR IN LEVEL 63 SAVE
003183  0D73  C380 124C                  LNJ    $B4,<RTC          SET RTC TO INTERRUPT
003184  0D75  0006                       DC     6                 AFTER 50 MSEC
003185  0D76  8F00 0EEE                  SAVE   <L63SV3,=X'7884'  PASS ALONG REGISTERS
        0D78  7884
003186  0D79  8E70 803F                  LEV    =Z'803F'          SUSPEND CURRENT AND GO TO 63
003187  0D7B  0F86                       B      >DOL2             RETURN HERE AFTER RTC RUPT
003188                            *
003189                            * LEVEL 63 CODE. OUTPUT A DATA SET SCAN CONTROL WORD
003190                            * WITH BIT 0 = 1 (SCAN), BIT 1 =1 (SET STATUS), BIT 2 = 0
003191                            * (INTERRUPT),<BIT 3 = 1 (START CHANNEL PROGRAM)
003192                            *
003193  0D7C  B380 0FCC           LV63TB LNJ    $B3,<SETLCT       OUTPUT DATA SCAN BYTE
003194  0D7E  0E09                       DC     <SCTSVA
003195  0D7F  0F02                NOER20 NOP    >DOL2             WAIT FOR RTC INTERRUPT
003196  0D80  0FFF                       B      >NOER20
003197                            *
003198                            * RETURN HERE VIA RTC INTERRUPT. THEN INPUT LCT STATUS
003199                            *
003200  0D81  0005                DOL2   RTCF                     TURN OFF REAL TIME CLOCK
003201  0D82  C380 084E                  LNJ    $B4,<ILCTST       INPUT LCI STATUS TO R5
003202  0D84  82D5                       LB     =$R5,=X'10'       GET DATA SET CHANGE BIT
        0D85  0010
```

```
003203  0D86  0503                    BBT     >$+2+$AF
003204  0D87  F380 125C                LNJ     $B7,<ERROR              DATA SET SCAN BIT NOT SET
003205                                *
003206                                * RESET LEVEL TO ZERO SO BLOCK READ WON'T CAUSE INTERRUPT.
003207                                *
003208  0D89  1C0D                    LDV     $R1,=X'D'               ADDRESS OF LEVEL
003209  0D8A  C380 0E0C                LNJ     $B4,<OUTLCT             OUTPUT 0 TO LEVEL
003210                                *
003211  0D8C  C380 0E25                LNJ     $B4,<RPVLU              READ P VALUE TO TEMP
003212  0D8E  E870 0204                LDR     $R6,=X'204'
003213  0D90  D956                    CMR     $R5,=$R6
003214  0D91  0903                    BE      >$+2+$AF
003215  0D92  F380 125C                LNJ     $B7,<ERROR              CHANNEL PROGRAM NOT STARTED BY
003216                                *                                SCAN WITH BIT 3 OF CHAN COMMAND = 1.
003217  0D94  0F80 0D50                B       <DSLOOP                 TEST NEXT MASK BIT
003218                                *
003219                                * TEST THAT DATA SET SCAN WILL CAUSE INTERRUPT
003220                                *
003221  0D96  C380 0E18                SCIT1A LNJ     $B4,<CHINTZ             INITIALIZE CHANNEL
003222  0D98  C380 1205                LNJ     $B4,<DLAY               DELAY 25 MS.
003223                                *
003224  0D9A  B3C0 0231                LNJ     $B3,SETLCT              SEND OUT LCT
003225  0D9C  0D11                    DC      <SCIEST
003226                                *
003227                                * OUTPUT LCT DATA
003228                                *
003229                                *       MASK = ALL ONES (F)
003230                                *
003231  0D9D  B380 0FCC                LNJ     $B3,<SETLCT             SET LCI
003232  0D9F  0DA1                    DC      <SCIT1                  TABLE ADDRESS
003233  0DA0  0F84                    B       >SCIT2
003234                                *
003235  0DA1  FF0F                    SCIT1   DC      =Z'FF0F'                RECEIVE MASK
003236  0DA2  FF2F                    DC      =Z'FF2F'                XMIT MASK
003237  0DA3  0000                    DC      0                       END OF TABLE
003238  0DA4  C380 0DF6                SCIT2   LNJ     $B4,<LEVRCH             SET LCI BYTES FOR LEV6 + REI. CHANNEL
003239  0DA6  9380 0F18                LNJ     $B1,<INTITZ             INITIALIZE INTERRUPTS
003240  0DA8  1C06                    LDV     $R1,=6
003241  0DA9  9F30 26A0                STR     $R1,<ILV,$R3            SET FOR LEVEL 6 IN LEVEL TABLE
003242  0DAB  C380 10DB                LNJ     $B4,<SETVEC             SET UP VECTOR FOR LEVEL
003243  0DAD  1CFF                    LDV     $R1,=-1
003244  0DAE  9F30 2680                STR     $R1,<INMB,$R3           SET FOR ONE RUPT EXPECTED
003245  0DB0  1C0F                    LDV     $R1,=X'F'               SET FLAG FO SCAN
003246  0DB1  9F00 14D1                STR     $R1,<SCFLG
003247                                *
003248                                * SEND OUT COMPLEMENT OF NORMAL STATUS
003249                                *
003250  0DB3  9800 14D0                LDR     $R1,<TEMPST
003251  0DB5  C380 0E0C                LNJ     $B4,<OUTLCT             OUT PUT VALUE
003252                                *
003253                                * SWITCH TO LEVEL 63
003254                                *
003255  0DB7  9B80 0DC4                LAB     $B1,<SCIT3
003256  0DB9  9F80 0EEC                STB     $B1,<L63SV2             SET P VECTOR IN LEVEL 63 SAVE
003257  0DBB  C380 124C                LNJ     $B4,<RTC                SET RTC TO INTERRUPT
003258  0DBD  0006                    DC      6                       AFTER 50 MS
003259  0DBE  8F00 0EEE                SAVE    <L63SV3,=X'7884'
        0DC0  7884
003260  0DC1  8E70 803F                LEV     =Z'803F'                SUSPEND CURRENT AND GO TO 63
003261  0DC3  0F8B                    B       >DOL3
003262                                *
003263                                * LEVEL 63 CODE. OUTPUT A DATA SCAN CONTROL WORD WITH
003264                                * BIT 0=1, SCAN; BIT 1=0 (SET STATUS) BIT 2=1 (INTERRUPT),
003265                                * BIT 3=0 (START CHANNEL PROGRAM).
003266  0DC4  9870 A008                SCIT3   LDR     $R1,=Z'A008'
003267  0DC6  3B02                    BEVN    $R3,>$+2                BRANCH IF RECEIVE
003268  0DC7  1E20                    ADV     $R1,=X'20'              MODIFY FOR XMIT
003269  0DC8  C380 0E0C                LNJ     $B4,<OUTLCT             OUTPUT BYTE
003270  0DCA  0F00 0DC4                WRUPT   NOP     <SCIT3
003271  0DCC  0F58                    NOP     >SCIT2
003272  0DCD  0FFD                    B       >WRUPT                  WAIT FOR INTERRUPT
003273                                *
003274                                * RETURN HERE VIA RTC INTERRUPT. CHECK LCI STATUS TO INSURE
003275                                * THAT DATA SET SCAN BIT IS RESET.
003276                                *
003277  0DCE  0005                    DOL3    RTCF                    TURN OFF REAL TIME CLOCK
003278  0DCF  0F81 000F                B       DOLBB                   TEMPORARY BYPASS **********
003279  0DD1  C3C0 0433                LNJ     $B4,DLAY                WAIT FOR CHANNEL PROGRAM TO FINISH
003280  0DD3  C380 084E                LNJ     $B4,<ILCTSI             INPUT LCI STATUS TO R5
003281  0DD5  82D5                    LB      =$R5,=X'10'             GET DATA SET CHANGE BIT
        0DD6  0010
003282  0DD7  0563                    BBF     >$+2+$AF
003283  0DD8  F380 125C                LNJ     $B7,<ERROR              DATA SET SCAN BIT IS SET WHEN
003284                                *                                IT SHOULDN'T BE.
003285  0DDA  82D5                    LB      =$R5,=X'4000'           GET INTERRUPT STATUS BIT
        0DDB  4000
003286  0DDC  0503                    BBT     >$+2+$AF
003287  0DDD  F380 125C                LNJ     $B7,<ERROR              DATA SET SCAN DIDN'T SET RUPT BIT
003288                                *
003289                                * RESET LEVEL IN LCT SO FOLLOWING BLOCK READ WILL NOT CAUSE AN INTERRUPT
003290                                *
003291  0DDF  1C0D                    DOLBB   LDV     $R1,=X'D'               ADDRESS OF LEVEL
003292  0DE0  3B02                    BEVN    $R3,>$+2                BRANCH IF RCV
003293  0DE1  1E20                    ADV     $R1,=X'20'
003294  0DE2  C380 0E0C                LNJ     $B4,<OUTLCT             OUTPUT 0 TO LEVEL
003295                                * CHECK P COUNTER TO INSURE THAT THE CHANNEL PROGRAM
003296                                * DIDN'T START.
003297                                *
003298  0DE4  C380 0E25                LNJ     $B4,<RPVLU              READ P VALUE TO TEMP
003299  0DE6  E870 0200                LDR     $R6,=X'200'
003300  0DE8  D956                    CMR     $R5,=$R6
003301  0DE9  0903                    BE      >$+2+$AF
003302  0DEA  F380 125C                LNJ     $B7,<ERROR              CHANNEL PROGRAM  STARTED
003303                                *                                BY DATA SET SCAN  WHEN IT
003304                                *                                SHOULD NOT HAVE.
003305                                *
003306                                * CHECK THAT INTERRUPT OCCURED
003307                                *
003308  0DEC  9380 0FB9                LNJ     $B1,<CHLV               CHECK INTERRUPT OCCURED ON CHANNEL
003309                                *
003310                                * TEST FOR NEXT CHANNEL
003311                                *
003312  0DEE  8AD3                    INC     =$R3
```

```
003313  0DEF  0F80 0CFA              B       <DSSTLP                 GO TO NEXT CHANNEL
003314                          *
003315                          * CHANNEL PROGRAM FOR SCAN TEST
003316                          *
003317  0DF1  01BD              SCNPRG DC      =Z'01BD'                NOP,<WAIT
003318  0DF2  3100                     DC      =Z'3100'                WAIT, WAIT
003319  0DF3  BD31                     DC      =Z'BD31'
003320  0DF4  00BD                     DC      =Z'00BD'
003321  0DF5  3100                     DC      =Z'3100'
003322                          *
003323                          * SUBROUTINES FOR INTERRUPT TESTS
003324  0DF6  CED5              LEVRCH SWB     $B4,=$B5                SAVE B4
003325  0DF7  8C51                     STS     =$R1                    GET S REG
003326  0DF8  9570 03C0                AND     $R1,=Z'03C0'            STRIP TO CPU ID
003327  0DFA  9F00 1548                STR     $R1,<RTCHN
003328  0DFC  9670 0006                XOR     $R1,=6                  PUT IN LEVEL 6
003329  0DFE  C800 1565                LDR     $R4,<CONT11             FUNCTION CODE FOR OUTPUT INT CONT
003330  0E00  C380 1173                LNJ     $B4,<CGSCH              MODIFY FOR CHANNEL
003331  0E02  8051                     IO      =$R1,=$R4               OUTPUT INTERRUPT CONTROL
        0E03  0054
003332  0E04  0703                     BIOT    >$+2+$AF
003333  0E05  F380 125C                LNJ     $B7,<ERROR              COMMAND WAS NAK'ED
003334  0E07  CED5                     SWB     $B4,=$B5                RESTORE B4
003335  0E08  8384                     JMP     $B4
003336                          * LCT CONTROL FOR SCAN TEST.
003337                          *
003338  0E09  D008              SCTSVA DC      =Z'D008'                RCV SCAN CONTROL
003339  0E0A  D028                     DC      =Z'D028'                =ZMIT SCAN CONTROL
003340  0E0B  0000                     DC      0
003341                          *
003342                          * OUTPUT LCT BYTE CONTAINED IN R1
003343  0E0C  CED5              OUTLCT SWB     $B4,=$B5
003344  0E0D  C800 1560                LDR     $R4,<CONT5              OUTPUT LCT BYTE FUNC CODE
003345  0E0F  C380 1173                LNJ     $B4,<CGSCH              MODIFY FOR FOR CHANNEL
003346  0E11  8051                     IO      =$R1,=$R4               OUTPUT BYTE
        0E12  0054
003347  0E13  0703                     BIOT    >$+2+$AF
003348  0E14  F380 125C                LNJ     $B7,<ERROR              COMMAND WAS NAK'ED
003349  0E16  CED5                     SWB     $B4,=$B5                RESTORE B4
003350  0E17  8384                     JMP     $B4
003351                          *
003352                          * ISSUE CHANNEL INITIALIZE
003353  0E18  CED5              CHINTZ SWB     $B4,=$B5
003354  0E19  C800 1562                LDR     $R4,<CONT7              CHANNEL CONTROL
003355  0E1B  C380 1173                LNJ     $B4,<CGSCH              MODIFY FOR CHANNEL
003356  0E1D  8070 8000                IO      =Z'8000',=$R4
        0E1F  0054
003357  0E20  0703                     BIOT    >$+2+$AF
003358  0E21  F380 125C                LNJ     $B7,<ERROR              COMMAND WAS NAKED
003359  0E23  CED5                     SWB     $B4,=$B5                RESTORE B5
003360  0E24  8384                     JMP     $B4
003361                          *
003362                          * READ P VALUE FROM RAM
003363                          *
003364  0E25  1C06              RPVLU  LDV     $R1,=6
003365  0E26  BF00 14CD                STR     $R3,<TPR
003366  0E28  3B03                     BEVN    $R3,>DOL4               BRANCH IF READ
003367  0E29  88D3                     DEC     =$R3                    GET READ CHANNEL
003368  0E2A  1C26                     LDV     $R1,=38
003369  0E2B  C853              DOL4   LDR     $R4,=$R3
003370  0E2C  4F20                     MLV     $R4,=X'20'              FORM START OF LCT AREA
003371  0E2D  9A54                     ADD     $R1,=$R4                FORM ADDRESS OF P COUNTER
003372  0E2E  9F00 0E34                STR     $R1,<PVLU1
003373  0E30  9380 1098                LNJ     $B1,<RDATA              DO BLOCK READ
003374  0E32  14CE                     DC      <TEMP                   TO HERE
003375  0E33  0002                     DC      2                       2 BYTES
003376  0E34  0000              PVLU1  DC      0                       RAM ADDRESS
003377  0E35  0000                     DC      0                       START AT EVEN CPU BYTE
003378                          *
003379  0E36  D800 14CE                LDR     $R5,<TEMP
003380  0E38  D570 FFFF                AND     $R5,=Z'FFFF'
003381  0E3A  DF00 14CE                STR     $R5,<TEMP
003382  0E3C  B800 14CD                LDR     $R3,<TPR                GET BACK TESTED CHANNEL
003383  0E3E  8384                     JMP     $B4
003384                          *----------------------------------------------------------
003385                          *
003386                          * 2ND INTERRUPT TEST.
003387                          *     DO BLOCK WRITES AND READS AND CHECK INTERRUPTS.
003388                          *     3 DEFERED INTERRUPTS ARE ACCUMULATED ON ALL CHANNELS
003389                          *      - LEVELS 5 - 62 ARE USED
003390                          *
003391                          *
003392  0E3F  B870 5420         SITST  LDR     $R3,=A'T '
003393  0E41  BF00 1581                STR     $R3,<ERMG+1
003394                                 CALL    ZV$T,ZV$TC,TSTT
        0E43  FBC0 0003
        0E45  D380 0000  X
        0E47  0F80
        0E48  1648
003395  0E49  8E70 0080                LEV     =Z'0080'                SUSPEND + QUICK CHANGE TO LEV 0
003396  0E4B  5C05                     LDV     $R5,=5                  STARTING LEVEL NUMBER
003397  0E4C  DF00 0E7E                STR     $R5,<LVSTR              STARTING LEVEL
003398                          *
003399  0E4E  BB80 1EBB        PRILP  LAB     $B3,<SDB                GET ADDRESS OF SEND DATA BUFFER
003400  0E50  BF80 0E64                STB     $B3,<ADD
003401  0E52  C870 0200                LDR     $R4,=X'200'             RAM ADDRESS
003402  0E54  CF00 0E65                STR     $R4,<RAD
003403  0E56  8753                     CL      =$R3                    R3 IS CHANNEL COUNTER
003404  0E57  2CF0                     LDV     $R2,=-16
003405  0E58  4C02                     LDV     $R4,=2                  RANGE OF TWO BYTES
003406                          *
003407                          * INITIALIZE CONTROLLER
003408                          *
003409  0E59  C380 10FC                LNJ     $B4,<GENITZ             MLCC GENERAL INITIALIZE
003410  0E5B  9380 0F18                LNJ     $B1,<INTITZ             INITIALIZE INTERRUPTS
003411                          *
003412                          * SET UP 3 CCB'S FOR EACH CHANNEL
003413                          *
003414  0E5D  C380 10EC        CCBLP  LNJ     $B4,<FLN                FIND ACTIVE LINE
003415  0E5F  0F92                     B       >RTI                    ALL HAVE BEEN FOUND
003416                          *
003417  0E60  8751                     CL      =$R1                    R1= 0 MEANS LEFT BYTE BOUNDRY
003418  0E61  5CFD                     LDV     $R5,=-3
```

```
003419                                  *
003420  0E62  C380 1189                 NEXT5   LNJ     $B4,<MCCB               FORM CCB
003421  0E64  14C7                      ADD     DC      <DUMMY
003422  0E65  0000                      RAD     DC      =X'0'                   RAM ADDRESS
003423                                  *
003424  0E66  57FC                              BINC    $R5,>NEXT5
003425  0E67  8AD3                              INC     =$R3                    SET FOR NEXT CHANNEL
003426  0E68  8A80 0E64                         INC     <ADD+$AF-1              BUMP CPU ADDRESS
003427  0E6A  8A80 0E65                         INC     <RAD                    BUMP RAM ADDRESS
003428  0E6C  8A80 0E65                         INC     <RAD                    BUMP AGAIN
003429  0E6E  AF48 FFF5                         STR     $R2,*ADD                SET UP DATA - ASCENDING PATTERN, -16 TO -1
003430  0E70  27ED                              BINC    $R2,>CCBLP              BRANCH IF MORE TO GO
003431                                  * SET UP LEVELS
003432  0E71  0800 0E7E                 RT1     LDR     $R5,<LVSTR              SET LEVEL START
003433  0E73  A870 8000                         LDR     $R2,=Z'8000'
003434  0E75  9852                      RT6     LDR     $R1,=$R2
003435  0E76  9500 14BD                         AND     $R1,<IMASK              STRIP TO CHANNEL BIT
003436  0E78  1900 0E86                         BEZ     $R1,<RT7A
003437                                  *
003438  0E7A  9F00 0E7F                         STR     $R1,<CHAN               INITIALIZE CHANNEL
003439  0E7C  9380 0FA8                         LNJ     $B1,<FTBL               SET LEVEL
003440  0E7E  0000                      LVSTR   DC      0                       LEVEL
003441  0E7F  0000                      CHAN    DC      0                       CHANNEL
003442  0E80  26A0                              DC      <ILV                    LEVEL TABLE
003443  0E81  8AD5                              INC     =$R5
003444  0E82  5D3F                              CMV     $R5,=63                 CHECK IF LAST LEVEL
003445  0E83  0905                              BE      >RT5                    BRANCH IF LAST LEVEL
003446  0E84  DF00 0E7E                         STR     $R5,<LVSTR              SET FOR NEXT LEVEL
003447  0E86  2041                      RT7A    SOR     $R2,1
003448  0E87  29EE                              BNEZ    $R2,>RT6                DO NEXT CHANNEL
003449                                  *
003450                                  *  FORM MASK FOR LINES TO BE TESTED
003451                                  *
003452  0E88  1CF0                      RT5     LDV     $R1,=-16
003453  0E89  8752                              CL      =$R2
003454                                  *
003455  0E8A  B810 26B0                 RT7     LDR     $R3,<ILV+16,$R1         GET LEVEL FOR CHANNEL
003456  0E8C  8AD1                              INC     =$R1
003457  0E8D  2011                              SCL     $R2,1
003458  0E8E  3903                              BEZ     $R3,>RT8
003459  0E8F  A670 0001                         XOR     $R2,=Z'0001'            PUT IN BIT FOR CHANNEL
003460  0E91  1879                      RT8     BLZ     $R1,>RT7                MORE TO GO
003461                                  *
003462  0E92  AF00 0E9B                         STR     $R2,<RT2
003463  0E94  AF00 0EA0                         STR     $R2,<RT3
003464  0E96  AF00 14E1                         STR     $R2,<MASK
003465                                  *
003466                                  *
003467                                  * SET UP INTERRUPT INFORMATION
003468                                  *
003469  0E98  9380 0FA8                         LNJ     $B1,<FTBL               SET EXPECTED INTERRUPT COUNT
003470  0E9A  FFFD                              DC      -3                      3 INTERRUPT/CHANNEL
003471  0E9B  0000                      RT2     DC      0                       MASK FOR TESTED CHANNELS
003472  0E9C  2680                              DC      <INMB                   TABLE ADDRESS
003473                                  *
003474  0E9D  9380 0FA8                         LNJ     $B1,<FTBL               SET EXPECTED STATUS
003475  0E9F  5000                              DC      =Z'5000'                WHAT STAT SHOULD BE
003476  0EA0  0000                      RT3     DC      0                       MASK FOR TESTED CHANNELS
003477  0EA1  2690                              DC      <IST                    STATUS TABLE
003478                                  *
003479                                  * SET INTERRUPT VECTORS
003480                                  *
003481  0EA2  C380 10DB                         LNJ     $B4,<SETVEC
003482                                  *
003483                                  * SEND OUT LEVEL AND CHANNEL
003484                                  *
003485  0EA4  B380 10CA                         LNJ     $B3,<SETV
003486  0EA6  8753                              CL      =$R3
003487  0EA7  C380 10EC                         LNJ     $B4,<FLN                FIND LINE NUMBER
003488  0EA9  0000                              HLT                             NO ACTIVE LINE NUMBERS
003489                                  *
003490                                  * TURN ON ALL CHANNELS
003491                                  *
003492  0EAA  1CFD                              LDV     $R1,=-3                 3 BLOCK OPERATIONS/CHAN
003493  0EAB  B380 10B0                 RT10    LNJ     $B3,<TRNON              TURN ON ALL CHANNELS TO BE TESTED
003494  0EAD  0400                              DC      =Z'0400'                START BLOCK WRITE CHANNEL CONTROL
003495  0EAE  0001                              DC      1                       BLOCK WRITE OPERATION
003496                                  *
003497  0EAF  B380 10B0                         LNJ     $B3,<TRNON              TURN ON CHANNELS
003498  0EB1  0800                              DC      =Z'0800'                BLOCK READ OPERATION
003499  0EB2  0002                              DC      2                       BLOCK READ
003500                                  *
003501  0EB3  C380 1205                         LNJ     $B4,<DLAY               DELAY 25 MS
003502  0EB5  1780 0EAB                         BINC    $R1,<RT10               BRANCH IF NOT LAST TIME
003503                                  *
003504                                  * SET UP RTC FOR 100 MILLISEC
003505                                  *
003506  0EB7  C380 124C                         LNJ     $B4,<RTC
003507  0EB9  000C                              DC      =12                     120'S OF A SEC
003508                                  * NOW SWITCH LEVELS TO 63 AND WAIT FOR SMOKE TO CLEAR
003509  0EBA  8F00 0EEE                         SAVE    <L63SV3,=X'7884'
              0EBC  7884
003510  0EBD  8E70 803F                         LEV     =Z'803F'                SUSPEND CURRENT AND DISPATCH 63
003511                                  *
003512                                  * RETURN HERE AFTER INTERRUPTS HAVE OCCURED
003513  0EBF  0005                              RTCF                            SHUT OFF RTC
003514                                  * CHECK ALL INTERRUPTS OCCURED
003515  0EC0  9380 0FB9                         LNJ     $B1,<CHLV               CHECK ALL LEVELS INTERRUPTED
003516                                  *
003517                                  * END OF 1 LOOP THRU
003518                                  *
003519  0EC2  5D3F                              CMV     $R5,=63                 CHECK FOR LAST LEVEL
003520  0EC3  0200 0E4E                         BL      <PRILP                  MORE TO GO
003521  0EC5  8382                              JMP     $B2                     EXIT TEST
003522                                  *
003523                                  * READ DATA SET STATUS
003524                                  *
003525  0EC6  8F40 0650                 DSSTA   SAVE    SAV3,=Z'0008'           SAVE B4
              0EC8  0008
003526  0EC9  C800 1566                         LDR     $R4,<CONT12             GET FUNCTION CODE
003527  0ECB  C380 1173                         LNJ     $B4,<CGSCH              OR IN CHANNEL NUMBER
003528  0ECD  8055                              IO      =$R5,=$R4               INPUT DATA SET STATUS
              0ECE  0054
```

```
003529  0ECF  0703                    BIOT    >$+2+$AF
003530  0ED0  F380 125C                LNJ     $B7,<ERROR              COMMAND WAS NAK'ED
003531  0ED2  8FC0 0644                RSTR    SAV3,=Z'0008'
        0ED4  0008
003532  0ED5  8384                    JMP     $B4                     DONE
003533                          *
003534                          *
003535                          *
003536                          *
003537                          *
003538                          * INTERRUPT SAVE AREA - LEV 4
003539                          *
003540  0ED6  0000              ISAV4   RESV    $AF,0                   TRAP SAVE POINTER
003541  0ED7  0000                      RESV    1,0                     CHAN, LEVEL
003542  0ED8  0000                      RESV    1,X'0'                  SAVE NO REGISTERS
003543  0ED9  0000                      RESV    1,0                     RFU
003544  0EDA  105C              IS4S    DC      <IMON                   CCB SCHEDULAR
003545  0EDB  FFFF                      DC      -1                      SET PRIVILEGE BIT
003546                          *
003547                          * INTERRUPT SAVE AREA - ERROR INTERRUPT
003548                          *
003549  0EDC  0000              ERRINT  RESV    $AF,0                   TRAP SAVE AREA POINTER
003550  0EDD  0000                      RESV    1,0                     DEVICE ID
003551  0EDE  0000                      RESV    1,0                     RFU
003552  0EDF  0000                      DC      Z'0000'                 SAVE
003553  0EE0  0EFE                      DC      <ERUPT                  ERROR RUPT ROUTINE
003554  0EE1  FFFF                      DC      -1                      SET PRIVILEGE
003555                          *
003556                          * RTC SAVE AREA
003557                          *
003558  0EE2  0000              RTCSAV  RESV    $AF,0
003559  0EE3  0000                      RESV    1,0                     DEVICE ID GETS STORED HERE
003560  0EE4  0000                      RESV    1,0                     RFU
003561  0EE5  0000                      DC      Z'0000'                 SAVE
003562  0EE6  0F08              RTCSV1  DC      <RTCHDL                 POINTS TO INTERRUPT ROUTINE
003563  0EE7  FFFF                      DC      -1                      SET PRIVILEGE BIT
003564                          *
003565                          * INTERRUPT SAVE AREA - LEVEL 63
003566                          *
003567  0EE8  0000              L63SV1  RESV    $AF,0                   TRAP SAVE POINTER
003568  0EE9  0000                      RESV    1,0                     CHAN,<LEVEL
003569  0EEA  7884                      RESV    1,X'7884'               SAVE R1,R2,R3,R4,I,B5
003570  0EEB  0000                      RESV    1,0                     RFU
003571  0EEC  0EF4              L63SV2  DC      <L63HDL                 POINTS TO INT. ROUTINE
003572  0EED  FFFF                      DC      -1                      SET PRIVILEGE BIT
003573  0EEE                    L63SV3  RESV    5+$AF                   SAVE AREA
003574                          *
003575                          *
003576                          *
003577                          * LEVEL 63 CONTAINS 'WAIT' LOOP
003578                          *
003579  0EF4  0F00 14C7         L63HDL  NOP     <DUMMY
003580  0EF6  0FFE                      B       >L63HDL
003581                          *
003582                          *  TRAP SAVE AREA IS AT LOC 2 - 9
003583                          *
003584                          *
003585                          * TRAP ERROR ROUTINE
003586                          *
003587  0EF7  8C00 0F05         TRPERR  STS     <LAST                   STORE STATUS
003588  0EF9  8E70 0080                 LEV     =X'0080'                ERROR, SHOULD BE NO TRAPS
003589  0EFB  F380 125C                 LNJ     $B7,<ERROR
003590  0EFD  0000                      HLT
003591                          *
003592                          * ERROR INTERRUPT
003593                          *
003594  0EFE  8C00 0F05         ERUPT   STS     <LAST                   STORE LAST STATUS
003595  0F00  8E70 0080                 LEV     =X'0080'                ERROR, BAD INTERRUPT OCCURED
003596  0F02  F380 125C                 LNJ     $B7,<ERROR
003597  0F04  0000                      HLT
003598  0F05  0000              LAST    DC      0                       LAST STATUS STORED HERE
003599                          *
003600                          * RTC INTERRUPT ROUTINE
003601                          *
003602  0F06  8E70 8000         SETNEW  LEV     =Z'8000'                SCHEDULE LEVEL = 0
003603  0F08  8A80 0F0C         RTCHDL  INC     <RTCFLG
003604  0F0A  83C0 FFFB                 JMP     SETNEW                  GO TO LEVEL 0
003605  0F0C  0000              RTCFLG  DC      0                       REAL TIME CLOCK FLAG
003606                          *
003607                          * INITIALIZATION TABLE FOR INTERRUPTS
003608                          *
003609  0F0D  0EE6              INTBL   DC      <RTCSV1                 RTC SAVE POINTER
003610  0F0E  0F08                      DC      <RTCHDL                 RTC RUPT POINTER
003611  0F0F  0EEC                      DC      <L63SV2                 LEVEL 63 P SAVE
003612  0F10  0EF4                      DC      <L63HDL                 LEVEL 63 RUPT HANDLER
003613  0F11  0EDA                      DC      <IS4S                   LEVEL 4 SAVE
003614  0F12  105C                      DC      <IMON
003615  0F13  0F17                      DC      <STKPTR                 PRIORITY STACK POINTER
003616  0F14  26B0                      DC      <PRIST
003617  0F15  0F16              INLLST  DC      <INLLST+$AF             LAST IN LIST
003618  0F16  26C0              PRIEND  DC      <PRIST+16
003619  0F17  26B0              STKPTR  DC      <PRIST
003620                          *
003621                          *
003622                          *
003623                          *
003624                          * INITIALIZATION FOR INTERRUPTS
003625                          *
003626                          *    ENTERED BY LNJ $B1,<INTITZ
003627                          *
003628  0F18  8F40 05CE         INTITZ  SAVE    SAV1,=Z'FFFF'           SAVE ALL
        0F1A  FFFF
003629  0F1B  DB80 0F0D                 LAB     $B5,<INTBL              GET ADDRESS OF LIST
003630  0F1D  ACF5              ITM1    LDB     $B2,*$B5                B2 GETS POINTER
003631  0F1E  DD80 0F15                 CMB     $B5,<INLLST
003632  0F20  0904                      BE      >INTNX                  BRANCH MEANS WE GOT END OF LIST
003633  0F21  BCF5                      LDB     $B3,*$B5                B3 GETS VECTOR
003634  0F22  BF82                      STB     $B3,$B2                 PUT IN VECTOR
003635  0F23  0FFA                      B       >ITM1                   DO NEXT IN LIST
003636                          *
003637  0F24  DB80 0EDD         INTNX   LAB     $B5,<ERRINT+$AF         GET ADDRESS OF ERROR SAVE
003638  0F26  AB80 0000       X         LAB     $B2,<ZHISAZ             WHERE TO START STORING IT
003639  0F28  1CC1                      LDV     $R1,=-63                63 VECTORS
```

```
003640  0F29  DFF2                ITM2   STB    $B5,+$B2
003641  0F2A  17FF                       BINC   $R1,>ITM2
003642                            *
003643                            * SET UP VECTORS FOR TRAPS
003644                            *
003645  0F2B  1CEC                       LDV    $R1,=-20
003646  0F2C  DB80 0000       X          LAB    $B5,<ZHPFR          CLEAR  TRAP HANDLER POINTERS
003647                            *
003648  0F2E  AB80 0000       X          LAB    $B2,<ZHISAZ
003649  0F30  DFE2                SVEC   STB    $B5,-$B2
003650  0F31  17FF                       BINC   $R1,>SVEC
003651  0F32  AB80 0000       X          LAB    $B2,<ZHTSA          GET LOCATION OF TRAP SAVE AREA
003652  0F34  AF80 0000       X          STB    $B2,<ZHNTSA
003653  0F36  AB80 0000       X          LAB    $B2,<ZHISAZ         GET RUPT POINTER BASE
003654  0F38  DB80 266B                  LAB    $B5,<TOPSAV         ADDRESS OF LEVEL 0 SAVE AREA
003655  0F3A  DF82                      STB    $B5,$B2
003656                            * PUT IN RTC, LEV 4 + 63 VECTORS
003657  0F3B  DB80 0EE3                  LAB    $B5,<RTCSAV+$AF     ADDRESS OF RTC SAVE AREA
003658  0F3D  DFC2 0003                  STB    $B5,$B2.3*$AF
003659  0F3F  DB80 0ED7                  LAB    $B5,<ISAV4+$AF      GET LOC OF RUPT SCHEDULAR SAVE
003660  0F41  DFC2 0004                  STB    $B5,$B2.4*$AF
003661  0F43  DB80 0EE9                  LAB    $B5,<L63SV1+$AF     GET ADDRESS OF LEV 63 SAVE AREA
003662  0F45  DFC2 003F                  STB    $B5,$B2.63*$AF
003663  0F47  8751                      CL     =$R1
003664  0F48  9F00 0F0C                  STR    $R1,<RTCFLG         CLEAR OUT RTC FLAG
003665  0F4A  9F00 14D2                  STR    $R1,<CCBFLG         RESET CCB FLAG
003666  0F4C  9F00 14D1                  STR    $R1,<SCFLG          RESET SCAN FLAG
003667  0F4E  9F00 0001       X          STR    $R1,<ZHIAFB+1
003668  0F50  9F00 0002       X          STR    $R1,<ZHIAFB+2
003669  0F52  9F00 0003       X          STR    $R1,<ZHIAFB+3
003670  0F54  9870 8000                  LDR    $R1,=Z'8000'        SET LEVEL FOR 0
003671  0F56  9F00 0000       X          STR    $R1,<ZHIAFB
003672                            * CLEAR OUT TRAP SAVE AREA
003673  0F58  DB80 14DB                  LAB    $B5,<TSA2           GET TRAP SAVE POINTER
003674  0F5A  DF80 0000       X          STB    $B5,<ZHTSA
003675                            *
003676                            * INITIALIZE STATUS TABLE
003677                            *
003678  0F5C  8700 14E2                  CL     <MASK1
003679                                   CALL   ZV$F,IST,MASK1,C16
        0F5E  FBC0 0003
        0F60  D380 0000       X
        0F62  0F80
        0F63  2690
        0F64  14E2
        0F65  14A9
003680                            *
003681                            * INITIALIZE LEVEL TABLE
003682                            *
003683                                   CALL   ZV$F,ILV,MASK1,C16
        0F66  FBC0 0003
        0F68  D380 0000       X
        0F6A  0F80
        0F6B  26A0
        0F6C  14E2
        0F6D  14A9
003684                            *
003685                            * INITIALIZE COUNT TABLE
003686                            *
003687  0F6E  8A80 14E2                  INC    <MASK1
003688                                   CALL   ZV$F,INMB,MASK1,C16
        0F70  FBC0 0003
        0F72  D380 0000       X
        0F74  0F80
        0F75  2680
        0F76  14E2
        0F77  14A9
003689                            *
003690                            * FILL PRIORITY LIST
003691                            *
003692                                   CALL   ZV$F,PRIST,ALLONE,C16
        0F78  FBC0 0003
        0F7A  D380 0000       X
        0F7C  0F80
        0F7D  26B0
        0F7E  14A8
        0F7F  14A9
003693                            *
003694                            *
003695                            * CREATE INTERRUPT SAVE AREAS
003696                            *
003697  0F80  8700 14E2                  CL     <MASK1
003698                                   CALL   ZV$F,IS1,MASK1,CNFILL FILL SDB WITH 0'S
        0F82  FBC0 0003
        0F84  D380 0000       X
        0F86  0F80
        0F87  25AA
        0F88  14E2
        0F89  0FA7
003699  0F8A  BB80 25AA                  LAB    $B3,<IS1            START OF ISA'S
003700  0F8C  8751                      CL     =$R1                R1 IS INDEX
003701                            *
003702  0F8D  9CF3                MISA   LDB    $B1,+$B3            BUMP PAST TSP
003703  0F8E  9873                       LDR    $R1,+$B3            BUMP PAST CHANNEL, DEVICE
003704  0F8F  C870 7884                  LDR    $R4,=X'7884'        MASK
003705  0F91  CF73                      STR    $R4,+$B3            MASK
003706  0F92  C873                      LDR    $R4,+$B3            BUMP PAST SECOND MASK
003707  0F93  9B80 1020                  LAB    $B1,<IHD1
003708  0F95  9FF3                      STB    $B1,+$B3            P
003709  0F96  4CFF                      LDV    $R4,=-1
003710  0F97  CF73                      STR    $R4,+$B3            S
003711                            *
003712  0F98  9CF3                       LDB    $B1,+$B3            BUMP FOR B IN SAVE
003713  0F99  BBC3 0005                  LAB    $B3,$B3.5
003714  0F9B  BD80 14CA                  CMB    $B3,<ENDSAV
003715  0F9D  09F0                       BNE    >MISA               BRANCH IF NOT DONE
003716                            *
003717                            *
003718                            * CREATE LEVEL 0 SAVE AREA
003719                            *
003720  0F9E  1CFF                MISB   LDV    $R1,=-1
003721  0F9F  9F00 266C                  STR    $R1,<TOPSAV+1
003722  0FA1  9F00 266E                  STR    $R1,<TOPSAV+2+$AF
```

```
003723  0FA3  8FC0 0543          RSTR   SAV1,=Z'FFFF'        RESTORE ALL REG
        0FA5  FFFF
003724  0FA6  8381              JMP    $B1
003725                      *
003726  0FA7  00D6          CNFILL DC     (157+57*$AF)
003727                      *
003728                      * ROUTINES TO SUPPORT INTERRUPT SERVICING
003729                      *
003730                      *******
003731                      *
003732                      ***
003733                      *
003734                      *  FILL 16 WORD DATA TABLE,
003735                      *  USED TO FILL ILV,IST,PRIST,NXRNG,CFIG,INMB
003736                      *
003737                      *  CALLED BY
003738                      *
003739                      *     LNJ  $B1,<FTBL
003740                      *     DC  XX  VALUE
003741                      *     DC  YY  CHANNELS
003742                      *     DC  <ZZ  TABLE ADDRESS
003743                      *
003744  0FA8  8F40 053E      FTBL   SAVE   SAV1,=Z'70B0'        R1,R2,R3,B2,B3,I
        0FAA  70B0
003745  0FAB  A871              LDR    $R2,+$B1             PICK UP NUMBER
003746  0FAC  B871              LDR    $R3,+$B1             PICK UP MASK
003747  0FAD  1CF0              LDV    $R1,=-16             SET COUNTER
003748  0FAE  ACF1              LDB    $B2,+$B1
003749  0FAF  BBC2 0010         LAB    $B3,$B2,16           ADDRESS OF END OF TABLE
003750  0FB1  3882          SLVLP  BGEZ   $R3,>ITM7         BRANCH IF BIT 0 = 0
003751  0FB2  AF13              STR    $R2,$B3,$R1
003752  0FB3  3011          ITM7   SCL    $R3,1                ROTATE R3
003753  0FB4  17FD              BINC   $R1,>SLVLP
003754  0FB5  8FC0 0531         RSTR   SAV1,=Z'70B0'
        0FB7  70B0
003755  0FB8  8381              JMP    $B1                  DONE
003756                      *
003757                      * CHECK EXPECTED NUMBER OF INTERRUPTS ON EACH CHANNEL
003758                      *
003759  0FB9  8F40 052D      CHLV   SAVE   SAV1,=Z'3000'
        0FBB  3000
003760  0FBC  B753              CL     =$R3                 WILL TRACK CHANNEL NUMBER
003761  0FBD  2CF0              LDV    $R2,=-16
003762  0FBE  89B0 26A0      CHLV1  CMZ    <ILV,$R3             CHECK IF CHANNEL WAS TO BE TESTED
003763  0FC0  0906              BE     >CHLV2
003764  0FC1  89B0 2680          CMZ    <INMB,$R3
003765  0FC3  0903              BE     >CHLV2
003766  0FC4  F380 125C          LNJ    $B7,<ERROR           WRONG NUMBER OF INTERRUPTS ON CHANNEL
003767  0FC6  8AD3          CHLV2  INC    =$R3                 BUMP CHANNEL NUMBER
003768  0FC7  27F7              BINC   $R2,>CHLV1
003769  0FC8  8FC0 051E         RSTR   SAV1,=Z'3000'
        0FCA  3000
003770  0FCB  8381              JMP    $B1                  RETURN
003771                      *
003772                      * SET UP LCT FOR CHANNEL SPECIFIED IN R3
003773                      *
003774                      *              LNJ $B3,<SETLCT
003775                      *               =DC  TABLE
003776                      *
003777  0FCC  8F40 052A      SETLCT SAVE   SAV2,=Z'E8E0'        R1,<R2,<R4,<B2,<B1
        0FCE  E8E0
003778  0FCF  9CF3              LDB    $B1,+$B3             GET ADDRESS OF TABLE
003779  0FD0  8751              CL     =$R1
003780                      *
003781  0FD1  A811          LCT4   LDR    $R2,$B1,$R1          GET BYTE TO OUTPUT
003782  0FD2  8AD1              INC    =$R1
003783  0FD3  2985              BNEZ   $R2,>LCT5            BRANCH IF NOT AT END OF TABLE
003784  0FD4  8FC0 0522          RSTR   SAV2,=Z'E8E0'
        0FD6  E8E0
003785  0FD7  8383              JMP    $B3                  RETURN
003786                      *
003787  0FD8  C800 1560      LCT5   LDR    $R4,<CONT5           FUNCTION CODE FOROUT LCT BYTE
003788  0FDA  C380 1173          LNJ    $B4,<CGSCH           FORM IO CONTROL WORD
003789  0FDC  8052          LCT3   IO     =$R2,=$R4            OUTPUT BYTE
        0FDD  0054
003790  0FDE  07FE              BIOF   >LCT3                CHECK IF TAKEN
003791  0FDF  0FF2              B      >LCT4                GET NEXT BYTE
003792                      *
003793                      *
003794                      *ENDLCT ROUTINE TO SEND LCT BYTES TO LEGAL LCT AREAS
003795                      *
003796  0FE0  8F40 0526      SNDLCT SAVE   SAVN,=Z'E8E0'        SAVE REGISTERS
        0FE2  E8E0
003797  0FE3  9CF3              LDB    $B1,+$B3             GET ADDRESS OF THE TABLE
003798  0FE4  9873              LDR    $R1,+$B3
003799                      *
003800  0FE5  8AD1          LCN4   INC    =$R1
003801  0FE6  9970 0000          CMR    $R1,=0               IS IT LCT 0
003802  0FE8  0901 FFFC          BE     LCN4                 DONT  SEND BYTE IF LCT 0
003803  0FEA  9970 0001          CMR    $R1,=1
003804  0FEC  0901 FFF8          BE     LCN4
003805  0FEE  9970 0005          CMR    $R1,=5
003806  0FF0  0901 FFF4          BE     LCN4
003807  0FF2  9970 0009          CMR    $R1,=9
003808  0FF4  0901 FFF0          BE     LCN4
003809  0FF6  9970 0015          CMR    $R1,=21
003810  0FF8  0901 FFEC          BE     LCN4
003811  0FFA  9970 0016          CMR    $R1,=22
003812  0FFC  0901 FFE8          BE     LCN4
003813  0FFE  9970 0020          CMR    $R1,=32
003814  1000  0901 FFE4          BE     LCN4
003815  1002  9970 0021          CMR    $R1,=33
003816  1004  0901 FFE0          BE     LCN4
003817  1006  9970 0025          CMR    $R1,=37
003818  1008  0901 FFDC          BE     LCN4
003819  100A  9970 0035          CMR    $R1,=53
003820  100C  0901 FFD8          BE     LCN4
003821  100E  9970 0036          CMR    $R1,=54
003822  1010  0901 FFD4          BE     LCN4
003823  1012  A811              LDR    $R2,$B1,$R1          GET THE BYTE
003824  1013  2985              BNEZ   $R2,>LCN5
003825  1014  8FC0 04F2          RSTR   SAVN,=Z'E8E0'
        1016  E8E0
```

```
003826  1017  8383                    JMP     $B3                     RETURN TO CALLER
003827                                *
003828  1018  C800 1560                LCN5    LDR     $R4,<CONT5      FUNCTION CODE FOR OUT LCT
003829  101A  C380 1173                        LNJ     $B4,<CGSCH      FORM IO CONTROL WORD
003830  101C  8052                    LCN3    IO      =$R2,=$R4
        101D  0054
003831  101E  07FE                            BIOF    >LCN3
003832  101F  0FC6                            B       >LCN4
003833                                *
003834                                *
003835                                ********************
003836                                *
003837                                *  INTERRUPT HANDLER ROUTINE
003838                                *
003839                                ********************
003840                                *
003841  1020  8C51                    IHD1    STS     =$R1            STORE STATUS IN R1
003842  1021  9570 003F                        AND     $R1,=X'3F'      STRIP TO LEVEL
003843  1023  B878 0000                        LDR     $R3,$IV,0       GET CHANNEL NUMB
003844  1025  B570 FC00                        AND     $R3,=Z'FC00'    STRIP TO MLCP ADDRESS
003845  1027  B900 14C3                        CMR     $R3,<MLCADD
003846  1029  0905                            BE      >IHDQ5          BRANCH IF GOOD
003847  102A  8E70 0080                        LEV     =X'0080'        QUICK CHANGE TO LEVEL 0
003848  102C  F380 125C                        LNJ     $B7,<ERROR      MLCP GAVE CP WRONG ADDRESS AFTER INTERRUPT
003849  102E  B878 0000                IHDQ5   LDR     $R3,$IV,0
003850  1030  B570 03C0                        AND     $R3,=X'3C0'     STRIP TO CHANNEL INFO
003851  1032  BF54                            STR     $R3,=$R4
003852  1033  3046                            SOR     $R3,6           GETS SUBCHAN IN R3
003853  1034  89B0 2680                        CMZ     <INMB,$R3       COMPARE NUMBER OF<INTERRUPTS
003854  1036  0985                            BNE     >IHDQ1
003855  1037  8E70 0080                        LEV     =X'0080'        SWITCH TO LEVEL 0
003856  1039  F380 125C                        LNJ     $B7,<ERROR      TOO MANY INTERRUPTS ON THIS CHANNEL
003857                                *
003858  103B  0801 0005                IHDQ1   BAL     IHDQ2           BRANCH IF NEG
003859  103D  8E70 0080                        LEV     =X'0080'        SWITCH TO LEVEL 0
003860  103F  F380 125C                        LNJ     $B7,<ERROR      ERROR, UNEXPECTED INTERRUPT
003861  1041  9930 26A0                IHDQ2   CMR     $R1,<ILV,$R3    COMPARE LEVELS
003862  1043  0905                            BE      >IHDQ3
003863  1044  8E70 0080                        LEV     =X'0080'        SWITCH TO LEVEL 0
003864  1046  F380 125C                        LNJ     $B7,<ERROR      WRONG CHANNEL INTERRUPTED THIS LEVEL
003865  1048  8980 14D1                IHDQ3   CMZ     <SCFLG          CHECK SCAN FLAG
003866  104A  098E                            BNE     >IHDQ4          IF SET, NO VALID CCB STATUS
003867  104B  C600 155F                        XOR     $R4,<CONT4      FORMS STATUS
003868  104D  8052                            IO      =$R2,=$R4       READ NEXT STATUS
        104E  0054
003869  104F  0F01 FFF8                        NOP     IHDQ3           DELAY
003870  1051  A930 2690                        CMR     $R2,<IST,$R3
003871  1053  0905                            BE      >IHDQ4
003872  1054  8E70 0080                        LEV     =X'0080'        SWITCH TO LEVEL 0
003873  1056  F380 125C                        LNJ     $B7,<ERROR      STATUS ERROR AFTER INTERRUPT
003874  1058  8E70 8004                IHDQ4   LEV     =Z'8004'        SUSPEND, SCAN, AND DISPATCH LEV 4
003875  105A  0F81 FFC5                        B       IHD1
003876                                ********************
003877                                *
003878                                * INTERRUPT MONITOR ROUTINE - LEVEL 4
003879                                *
003880                                ********************
003881  105C  DC80 0F17                IMON    LDB     $B5,<STKPTR     GET PRIORITY STACK POINTER
003882  105E  DD80 0F16                        CMB     $B5,<PRIEND     CHECK IF AT END
003883  1060  0284                            BGE     >IMON2          STACK IS FULL
003884                                *
003885  1061  BF75                            STR     $R3,+$B5        STORE CHANNEL ON STACK
003886  1062  DF80 0F17                        STB     $B5,<STKPTR     STORE NEW STACK POINTER
003887                                *
003888  1064  8AB0 2680                IMON2   INC     <INMB,$R3       INCREMENT INTERRUPT COUNT
003889                                *
003890  1066  8E70 803F                IMON4   LEV     =Z'803F'        SCHEDULE LEV 03 AND SUSPEND
003891  1068  0FF4                            B       >IMON           PICK UP NEXT TIME FROM HERE
003892                                *
003893                                * BLOCK WRITE DATA TO RAM
003894                                *
003895                                *       LNJ     $B1,<SDATA
003896                                *       DC      DATA            LOCATION OF DATA
003897                                *       DC      RANGE           NUMBER OF DATA BYTES
003898                                *       DC      RAMAD           RAM ADDRESS
003899                                *       DC      EVEN            0 = EVEN BYTE CPU ADDRESS
003900                                *                                   BIT 15 = 1 FOR ODD BYTE START
003901                                *                                   BIT 0 = 0 FOR 25 MS DELAY BEFORE RETURNIN
003902                                *                                   BIT 0 = 1 FOR 250 MS DELAY BEFORE RETURNI
003903                                *
003904                                * R3 MUST CONTAIN THE CHANNEL NUMBER
003905                                *
003906  1069  8F40 048D                SDATA   SAVE    SAV2,=Z'FFBF'   SAVE ALL BUT B1
        106A  FFBF
003907  106C  DCF1                            LDB     $B5,+$B1        GET ADDRESS OF DATA
003908  106D  C871                            LDR     $R4,+$B1        GET RANGE
003909  106E  DF80 1079                        STB     $B5,<SPRG5
003910  1070  A871                            LDR     $R2,+$B1        GET RAM ADDRESS
003911  1071  AF00 107A                        STR     $R2,<SPRG2
003912  1073  9871                            LDR     $R1,+$B1        LOAD START BYTE INDEX
003913  1074  9F57                            STR     $R1,=$R7        STORE  BYTE INDEX
003914  1075  9570 7FFF                        AND     $R1,=X'7FFF'
003915  1077  C380 1189                        LNJ     $B4,<MCCB       FORM CCB
003916  1079  0000                SPRG5   RESV    $AF,0           CPU ADDRESS
003917  107A  0000                SPRG2   DC      0               RAM ADDRESS
003918  107B  C380 1177                        LNJ     $B4,<CHCT       DO CHANNEL CONTROL
003919  107D  0F82                            B       >$+2
003920  107E  0400                            DC      Z'0400'         BLOCK WRITE
003921                                *
003922                                * PROGRAM ARIVES HERE FROM SDATA OR RDATA AS WELL AS SPRG.
003923                                *
003924  107F  7880 1086                SPRG3   BGEZ    $R7,<SPRG7
003925  1081  C380 11C8                        LNJ     $B4,<TEST       WAIT FOR STATUS COMPLETE
003926  1083  0F82                            B       >$+2
003927  1084  001E                            DC      30              250 MS TIMEOUT
003928  1085  0F85                            B       >SPRG8
003929  1086  C380 1215                SPRG7   LNJ     $B4,<DLAYLG     DELAY 25 MS
003930  1088  C380 1161                        LNJ     $B4,<INXT       INPUT NEXT STATUS
003931  108A  82D5                SPRG8   LB      =$R5,=X'1000'   GET STATUS COMPLETE BIT
        108B  1000
003932  108C  0503                            BBT     >$+2+$AF
003933  108D  F380 125C                        LNJ     $B7,<ERROR      STATUS COMPLETE NOT SET AFTER BLOCK WRITE
003934  108F  82D5                            LB      =$R5,=7
```

```
                1090  0007
003935          1091  0583                    BBF     >$+2+$AF
003936          1092  F380 125C               LNJ     $B7,<ERROR          ERROR, PARITY,MEMORY,OR RESOURCES
003937          1094  8F80 14F7               RSTR    <SAV2,=Z'FFBF'
                1096  FFBF
003938          1097  8381                    JMP     $B1
003939                                *****
003940                                *
003941                                * BLOCK READ FROM RAM.- CHAN. NUMBER MUST BE IN R3.
003942                                *
003943                                *       LNJ     $B1,<RDATA
003944                                *       DC      INBUFF              INPUT BUFFER ADDRESS
003945                                *       DC      RANGE               NUMBER OF BYTES
003946                                *       DC      RAMAD               RAM ADDRESS
003947                                *       DC      EVEN                0 = EVEN BYTE CPU ADDRESS
003948                                *                                       BIT 15 = 15 FOR ODD BYTE ADDRESS
003949                                *                                       BIT 0 = 1 FOR 250 MS DELAY AFTER STARTING
003950                                *                                       BIT 0 = 0 FOR 25 MS DELAY AFTER STARTING
003951                                *
003952          1098  8F40 045E       RDATA   SAVE    SAV2,=Z'FFBF'       SAVE EVERYTHING BUT B1.
                109A  FFBF
003953          109B  DCF1                      LDB     $B5,+$B1            GET IN BUFF ADD
003954          109C  DF80 10A8                 STB     $B5,<RDTA1
003955          109E  C871                      LDR     $R4,+$B1            GET RANGE IN BYTES
003956          109F  9871                      LDR     $R1,+$B1
003957          10A0  9F00 10A9                 STR     $R1,<RDTA2
003958          10A2  9871                      LDR     $R1,+$B1            PICK UP EVEN, ODD FLAG
003959          10A3  9F57                      STR     $R1,=$R7            STORE BYTE  INDEX
003960          10A4  9570 7FFF                 AND     $R1,=X'7FFF'
003961          10A6  C380 1189                 LNJ     $B4,<MCCB           FORM CCB
003962          10A8  14C7              RDTA1   DC      <DUMMY              CPU ADDRESS
003963          10A9  0000              RDTA2   DC      0                   RAM ADDRESS
003964          10AA  C380 1177                 LNJ     $B4,<CHCT           DO CHANNEL CONTROL
003965          10AC  0F82                      B       >$+2
003966          10AD  0800                      DC      Z'0800'             BLOCK READ
003967                                *
003968          10AE  0F81 FFD0                 B       SPRG3               EXIT
003969                                *
003970                                * TURN ON ALL CHANNELS TO BE TESTED
003971                                *
003972                                *     LNJ $B3,<TRNON
003973                                *     DC CONT CHAN CONTROL WORD
003974                                *     DC XX 1= XMIT ONLY, 2= REC. ONLY, 0= BOTH
003975                                *
003976          10B0  8F40 0436         TRNON   SAVE    SAV1,=Z'1F00'       R3,4,5,6
                10B2  1F00
003977          10B3  D873                      LDR     $R5,+$B3            GET CHANNEL CONTROL WORD
003978          10B4  E873                      LDR     $R6,+$B3            GET TYPE FLAG
003979          10B5  3C10                      LDV     $R3,=16             R3 GETS CHANNEL NUMBERS
003980          10B6  3705              ITM11   BDEC    $R3,>TALL           BRANCH IF NOT DONE
003981          10B7  8FC0 042F                 RSTR    SAV1,=Z'1F00'
                10B9  1F00
003982          10BA  8383                      JMP     $B3                 ALL DONE, EXIT
003983                                *
003984          10BB  C800 1562         TALL    LDR     $R4,<CONT7
003985          10BD  89B0 26A0                 CMZ     <ILV,$R3            CHECK IF LEVEL SET UP
003986          10BF  0977                      BE      >ITM11              BRANCH IF NOT SET UP
003987          10C0  6904                      BEZ     $R6,>ITM11A         CHECK TYPE OF XFER
003988          10C1  F856                      LDR     $R7,=$R6
003989          10C2  F653                      XOR     $R7,=$R3
003990          10C3  7BF3                      BODD    $R7,>ITM11          BRANCH IF NOT RIGHT TYPE
003991                                *
003992          10C4  C380 1173         ITM11A  LNJ     $B4,<CGSCH          FORM IO CONTROL WORD
003993          10C6  8055              ITM12   IO      =$R5,=$R4           START IO
                10C7  0054
003994          10C8  07FE                      BIOF    >ITM12              WAIT
003995          10C9  0FED                      B       >ITM11              DO NEXT
003996                                *
003997                                * SET LEVEL + CHANNEL FOR ALL ACTIVE CHANNELS
003998                                *
003999          10CA  8753              SETV    CL      =$R3
004000          10CB  9830 26A0         SETV2   LDR     $R1,<ILV,$R3        GET LEVEL
004001          10CD  190A                      BEZ     $R1,>SETV1
004002          10CE  C800 1565                 LDR     $R4,<CONT11         GET FUNCTION CODE
004003          10D0  C3C0 00A2                 LNJ     $B4,CGSCH           OR IN CHANNEL
004004          10D2  8051                      IO      =$R1,=$R4
                10D3  0054
004005          10D4  0703                      BIOT    >$+2+$AF
004006          10D5  F380 125C                 LNJ     $B7,<ERROR          COMMAND NEVER SHOULD BE NAKED
004007                                *
004008          10D7  8AD3              SETV1   INC     =$R3
004009          10D8  3D10                      CMV     $R3,=16
004010          10D9  09F2                      BNE     >SETV2
004011          10DA  8383                      JMP     $B3
004012                                *
004013                                * SET UP INTERRUPT VECTORS FOR ENTRIES IN LEVEL TABLE
004014                                *
004015          10DB  8752              SETVEC  CL      =$R2
004016          10DC  0B80 25AB                 LAB     $B5,<IS1+$AF        GET 1ST ISA SAVE POINTER
004017          10DE  0F04              SET1    NOP     >$+4
004018          10DF  BBC5 000C                 LAB     $B3,$B5,9+3*$AF     FORM ADDRESS OF NEXT ISA
004019          10E1  9820 26A0         SAF4    LDR     $R1,<ILV,$R2        GET LEVEL
004020          10E3  89D1              SAF5    CMZ     =$R1
004021          10E4  0903                      BE      >SET2               BRANCH IF NO LEVEL SET FOR CHANNEL
004022          10E5  DF90 0000       X                 STB     $B5,<ZH1SAZ,$<1     STORE VECTOR
004023          10E7  8AD2              SET2    INC     =$R2                SET FOR NEXT CHANNEL
004024          10E8  BFD5                      STB     $B3,=$B5
004025          10E9  2D11                      CMV     $R2,=17             CHECK FOR END
004026          10EA  0274                      BL      >SET1               MORE
004027          10EB  8384                      JMP     $B4                 EXIT
004028                                *
004029                                *
004030                                *           FIND FIRST ACTIVE CHANNEL
004031                                *
004032                                *             $R3-START CHECK ON THIS CHANNEL AND
004033                                *                 CYCLE THROUGH ALL CHANNEL IF
004034                                *                 ONE FOUND ON CHANNEL NUMBER IS HERE
004035                                *
004036                                *             LNJ  $B4,<FLN
004037                                *             <RETURN> IF NO LINES ON
004038                                *             <RETURN>
004039                                *
004040                                *
```

```
004041  10EC  8F40 03FA          FLN     SAVE    SAV1,=Z'A0A0'          SAVE REG. $R2,$B2
        10EE  A0A0
004042  10EF  2CFF                        LDV     $R2,=-1
004043  10F0  AB80 14AD                    LAB     $B2,<ATLT              ADDRESS OF ACTIVE LINE TABLE
004044  10F2  3D10                FLN2    CMV     $R3,=16                SEE IF AT BOTTOM OF TABLE
004045  10F3  0285                        BGE     >FLN3
004046  10F4  A97E                        CMR     $R2,$B2.+$R3           SEE IF INDEX SCH. ON
004047  10F5  097D                        BE      >FLN2
004048  10F6  88D3                        DEC     =$R3                   SET $R3 BACK TO VALUE FOUND
004049  10F7  A874                        LDR     $R2,+$B4               INCREMENT TO VALID RETURN
004050  10F8  8FC0 03EE          FLN3    RSTR    SAV1,=Z'A0A0'          RESTORE REG.
        10FA  A0A0
004051  10FB  8384                        JMP     $B4                    JUMP OUT OF ROUTINE
004052                          *
004053                          * GENERAL INITIALIZE FOR MLCC
004054                          *
004055  10FC  8F40 041A          GENITZ  SAVE    SAV3,=Z'1808'          SAVE B4,R4
        10FE  1808
004056  10FF  8753                        CL      =$R3
004057  1100  C380 10EC                    LNJ     $B4,<FLN               FIND ACTIVE LINE
004058  1102  0000                        HLT                            NO ACTIVE LINES
004059  1103  C800 1563                    LDR     $R4,<CONT9             GET FUNCTION CODE FOR INITIALIZE
004060  1105  C380 1173                    LNJ     $B4,<CGSCH             MODIFY FOR CHANNEL
004061  1107  8070 8000                    IO      =Z'8000',=$R4          INITIALIZE
        1109  0054
004062  110A  0703                        BIOT    >$+2+$AF
004063  110B  F380 125C                    LNJ     $B7,<ERROR             INITIALIZE WAS NAK'ED
004064  110D  C380 1215                    LNJ     $B4,<DLAYLG            WAIT 600 MS
004065  110F  8FC0 0407                    RSTR    SAV3,=Z'1808'
        1111  1808
004066  1112  8384                        JMP     $B4                    RETURN
004067                          *
004068                          *
004069                          *
004070                          *
004071                          *  PSEUDO - RANDOM DATA GENERATION ROUTINE
004072                          *           DATA IS BASED ON CRC GENERATOR
004073                          *
004074                          *      2 MODES OF OPERATION.
004075                          * *           INPUT - TAKES INITIAL CHAR AS BASE AND FORMS
004076                          *             TABLE OF DATA BASED ON CRC SPECIFIED.
004077                          *
004078                          *             OUTPUT - TAKES TABLE OF DATA AND FORMS ACCUMULATED CRC.
004079                          *EE    ONLY FOR CRC16 AND CCITT.
004080                          *
004081                          *      TO USE, MUST SET:
004082                          *             R6 - NUMBER OF BITS IN CHAR.
004083                          *             R7 - RANGE IN BYTES
004084                          *             B3 - TABLE ADDRESS
004085                          *             B6 - ADDRESS OF CRC TYPE.
004086                          *
004087                          *      FOR INPUT SET:
004088                          *
004089                          *             R5 - INITIAL CHAR. TO GIVE TO CRC GEN.
004090                          *
004091                          *      FOR OUTPUT SET:
004092                          *
004093                          *             R5 - MUST BE SET NEGATIVE.
004094                          *            CRC CHECK AND FILL ROUTINE
004095                          *
004096                          *
004097                          *                LNJ     $B4,<CRC
004098                          *
004099                          *
004100                          *      $B3 - ADDRESS WHERE CRC ACCUMULATOR IS STORED
004101                          *            OR CHARACTER IS LOADED IF $R5<0
004102                          *      $B6 - ADDRESS OF CRC TYPE
004103                          *
004104                          *
004105                          *      $R5 - IF >=0 CONTAINS INITIAL CHARACTER FOR FILLING TABLE
004106                          *            IF <0 NOT USED EXCEPT AS A FLAG FOR OUTPUT
004107                          *      $R6 - CONTAINS NUMBER OF BITS IN CHARACTER
004108                          *      $R7 - THE NUMBER OF CRC TO BE
004109                          *            GENERATED INTO/FROM THE TABLE AT $B3
004110                          *
004111                          *         THE ACCUMULATED CRC IS STORED IN LOCATION CRCAC.
004112                          *
004113                          *
004114  1113  8F40 0413          CRC     SAVE    SAV4,=Z'F994'
        1115  F994
004115  1116  C806                        LDR     $R4,$B6                PUT CRC TYPE INTO $R4
004116  1117  8751                        CL      =$R1
004117  1118  8700 14CE                    CL      <TEMP
004118  111A  8700 14CF                    CL      <TEMP1
004119  111C  8753                        CL      =$R3                   CLEAR $R3 FOR LONG SHIFT
004120  111D  8752                        CL      =$R2
004121  111E  8256                        NEG     =$R6
004122  111F  EF00 14DA                    STR     $R6,<IMGA+($AF-1)      STORE COUNTER
004123  1121  5885                        BGEZ    $R5,>CRCAA1            SEE IF OUTPUT OR INPUT
004124  1122  D0FF                        LDH     $R5,$B3.+$R3
004125  1123  8A80 14CE                    INC     <TEMP
004126  1125  7001                        SOL     $R7,1
004127  1126  8257              CRCAA1    NEG     =$R7
004128  1127  89D4              CRCAA     CMZ     =$R4                   IF LRC8
004129  1128  0901 0035                   BE      CRCF                   BRANCH TO CRCF
004130  112A  C970 F010                   CMR     $R4,=Z'F010'           IF CRC12
004131  112C  091A                        BE      >CRCD                  BRANCH TO CRCD
004132  112D  DF51              CRCA    STR     $R5,=$R1
004133  112E  5041                        SOR     $R5,1
004134  112F  9652                        XOR     $R1,=$R2
004135  1130  2041                        SOR     $R2,1
004136  1131  1B02                        BEVN    $R1,>CRCB
004137  1132  A654                        XOR     $R2,=$R4
004138  1133  67FA              CRCB    BINC    $R6,>CRCA
004139  1134  89C0 0399                   CMZ     TEMP
004140  1136  098E                        BNE     >CRCDD
004141  1137  D852                        LDR     $R5,=$R2
004142  1138  5048                        SOR     $R5,8
004143  1139  88D5                        DEC     =$R5
004144  113A  AF73              CRCC    STR     $R2,+$B3               STORE CRC ACCUM. IN TABLE
004145  113B  E800 14DA          CRCC2   LDR     $R6,<IMGA+($AF-1)      RESET CHAR. COUNT
004146  113D  77EA                        BINC    $R7,>CRCAA             DO NEW CHARACTER
004147  113E  AF00 14D3                    STR     $R2,<CRCAC             STORE LAST CRC ACCUMULATED
```

```
004148  1140  8FC0 03E6              RSTR   SAV4,=Z'F994'        RESTORE REGS.
        1142  F994
004149  1143  8384                   JMP    $B4
004150                         *
004151  1144  D0FF             CRCDD LDH    $R5,$B3,+$R3
004152  1145  0FF6                   B      >CRCC2
004153                         *
004154  1146  5004             CRCD  SOL    $R5,4
004155  1147  DF51                   STR    $R5,=$R1
004156  1148  5045                   SOR    $R5,5
004157  1149  9652                   XOR    $R1,=$R2
004158  114A  2045                   SOR    $R2,5
004159  114B  2004                   SOL    $R2,4
004160  114C  82D1                   LB     =$R1,=X'10'
        114D  0010
004161  114E  0582                   BBF    >CRCE
004162  114F  A654                   XOR    $R2,=$R4
004163  1150  67F6             CRCE  BINC   $R6,>CRCD
004164  1151  30CA                   DOR    $R3,10               FORM GENERATED CRC
004165  1152  3042                   SOR    $R3,2                ACCUMULATOR
004166  1153  2002                   SOL    $R2,2
004167  1154  3088                   DOL    $R3,8
004168  1155  AF73                   STR    $R2,+$B3
004169  1156  D852                   LDR    $R5,=$R2
004170  1157  30C8                   DOR    $R3,8
004171  1158  3002                   SOL    $R3,2
004172  1159  2042                   SOR    $R2,2
004173  115A  308A                   DOL    $R3,10
004174  115B  5048                   SOR    $R5,8
004175  115C  88D5                   DEC    =$R5
004176  115D  0FDE                   B      >CRCC2
004177                         *
004178  115E  A655             CRCF  XOR    $R2,=$R5
004179  115F  8AD5                   INC    =$R5
004180  1160  0FDA                   B      >CRCC
004181                         *
004182                         *
004183                         * INPUT NEXT STATUS TO R5
004184                         *
004185  1161  8F00 1535        INXT  SAVE   <SAV6,=Z'0008'       B4
        1163  0008
004186  1164  C800 155F              LDR    $R4,<CONT4           GET CONTROL WORD FOR INPUT NEXT STATUS
004187  1166  C380 1173              LNJ    $B4,<CGSCH           MODIFY FOR CHANNEL
004188  1168  8055                   IO     =$R5,=$R4            GET NEXT STATUS
        1169  0054
004189  116A  0703                   BIOT   >$+2+$AF
004190  116B  F380 125C              LNJ    $B7,<ERROR           INPUT NEXT STATUS WAS NAK'ED
004191  116D  DF00 154D              STR    $R5,<STAT            STORE LAST STATUS READ
004192  116F  8F80 1535              RSTR   <SAV6,=Z'0008'
        1171  0008
004193  1172  8384                   JMP    $B4
004194                         *
004195                         *
004196                         *
004197                         *
004198                         *
004199                         *                CHANGE CHANNEL
004200                         *
004201                         *      $R3 -CONTAINS CHANNEL WANTED
004202                         *      $R4 -CONTAINS I/O CONTROL WORD TO BE CHANGED
004203                         *
004204                         *      LNJ     $B4,<CGSCH
004205                         *
004206                         *
004207  1173  3006             CGSCH SOL    $R3,6                SHIFT IO CHANNEL POSITION
004208  1174  C453                   OR     $R4,=$R3             OR CHANNEL NUMBER INTO CONTROLWORD
004209  1175  3046                   SOR    $R3,6                SHIFT IO NORMAL POSITION
004210  1176  8384                   JMP    $B4                  GO BACK
004211                         *
004212                         * OUTPUT CHANNEL CONTROL
004213                         *
004214                         *   LNJ  $B4,CHCT
004215                         *   B  >$+2          RETURN
004216                         *   DC   XX          XX = CHANNEL CONTROL
004217                         *
004218  1177  8F00 14E7        CHCT  SAVE   <SAV1,=Z'0C0D'       R4,R5,B5,B7,B4
        1179  0C0D
004219  117A  D874                   LDR    $R5,+$B4             DUMMY
004220  117B  D874                   LDR    $R5,+$B4             GET CONTROL WORD
004221  117C  C800 1562              LDR    $R4,<CONT7           FUN CODE FOR CCB CONTROL
004222  117E  C380 1173              LNJ    $B4,<CGSCH           FORM IO CONTROL WORD
004223  1180  8055                   IO     =$R5,=$R4            OUTPUT CCB CONTROL
        1181  0054
004224  1182  0703                   BIOT   >CHZ
004225  1183  F380 125C              LNJ    $B7,<ERROR           ERROR, IO WAS NAK'ED
004226                         *
004227  1185  8F80 14E7        CHZ   RSTR   <SAV1,=Z'0C0D'
        1187  0C0D
004228  1188  8384                   JMP    $B4
004229                         *
004230                         *
004231                         *
004232                         *
004233                         *       CCB FORMATION
004234                         *
004235                         *         $R1 - CONTAINS INDEX OF CPU ADDRESS
004236                         *         $R3 - CONTAINS CHANNEL WANTED
004237                         *         $R4 - CONTAINS RANGE (NUMBER OF BYTES)
004238                         *
004239                         *         LNJ      $B4,<MCCB
004240                         *         DC       CPU ADDRESS
004241                         *         DC       RAM ADDRESS NUMBER OR CHANNEL CONTROL WORD.
004242                         *
004243                         *
004244  1189  8F40 035D        MCCB  SAVE   SAV1,=Z'FDF4'        SAVES $B1,$B3,$B2,$B5,$R7,$R5,$R4,R2,& $R1
        118B  FDF4
004245  118C  ACF4                   LDB    $B2,+$B4             LOAD $B2 WITH CPU ADDRESS
004246  118D  A874                   LDR    $R2,+$B4             PUT RAM ADDRESS IN $R2
004247  118E  DED4                   SWB    $B5,=$B4             ALLOW $B4 TO BE USE IN SUBR. CALL
004248  118F  D854                   LDR    $R5,=$R4
004249  1190  C800 155C              LDR    $R4,<CONT1           LOAD $R4 WITH I/O CONTROL WORD
004250  1192  C380 1173              LNJ    $B4,<CGSCH
004251  1194  8192                   IOLD   $B2,$R1,=$R4,=$R5    OUTPUT ADDRESS AND RANGE
```

```
                1195  0054
                1196  0055
004252          1197  0703                        BIOT    >$+2+$AF
004253          1198  F380 125C                   LNJ     $B7,<ERROR          IOLD WAS NAK'ED
004254          119A  C800 155E                   LDR     $R4,<CONT3          LOAD $R4 WITH I/O CONTROL WORD
004255          119C  C380 1173                   LNJ     $B4,<CGSCH          PUT I/O CONTROL WORD IN $R4
004256          119E  8052                        IO      =$R2,=$R4           OUTPUT MLCC RAM ADDRESS
                119F  0054
004257          11A0  0703                        BIOT    >$+2+$AF
004258          11A1  F380 125C                   LNJ     $B7,<ERROR          OUTPUT CONTROL FLAG WAS NAK'ED
004259          11A3  CED5                        SWB     $B4,=$B5            SWAP FOR SUBR. RETURN
004260          11A4  8FC0 0342                   RSTR    SAV1,=Z'FDF4'       RESTORE REGS.
                11A6  FDF4
004261          11A7  8384                        JMP     $B4
004262                                    *
004263                                    * INPUT LCT BYTE
004264                                    *
004265          11A8  DED4                INBYTE  SWB     $B5,=$B4
004266          11A9  C800 1564                   LDR     $R4,<CONT10         GET FUNCTION CODE
004267          11AB  C380 1173                   LNJ     $B4,<CGSCH          PUT IN CHANNEL NUMBER
004268          11AD  8055                        IO      =$R5,=$R4           INPUT
                11AE  0054
004269          11AF  0703                        BIOT    >$+2+$AF
004270          11B0  F380 125C                   LNJ     $B7,<ERROR          IO WAS NAK'ED
004271          11B2  D570 FF00                   AND     $R5,=Z'FF00'
004272          11B4  DED4                        SWB     $B5,=$B4
004273          11B5  8384                        JMP     $B4
004274                                    *
004275                                    *
004276                                    *
004277                                    *       PACK RAM INSTRUCTIONS
004278                                    *
004279                                    *
004280                                    *       $R5 - RANGE IN WORDS OF BUFFER WHERE PACKED INSTRUCTIONS GO.
004281                                    *
004282                                    *         LNJ     $B4,<PKRIN
004283                                    *         DC      ADDRESS OF UNPACKED INSTRUCTION
004284                                    *         DC      ADDRESS OF PACKED INSTRUCTIONS
004285                                    *
004286                                    *
004287          11B6  8F40 0330           PKRIN   SAVE    SAV1,=Z'CCD0'       SAVE $R1,$R4,$R5,$B3,$B1
                11B8  CCD0
004288          11B9  8255                        NEG     =$R5
004289          11BA  BCF4                        LDB     $B3,+$B4            LOAD $B3 WITH ADDRESSS OF UNPACKED INSTS.
004290          11BB  9CF4                        LDB     $B1,+$B4            LOAD $B4 WITH ADDRESS OF PACKED INSTS.
004291          11BC  C873                PKR1    LDR     $R4,+$B3            GET FIRST UNPACKED INST. IN $R4
004292          11BD  4008                        SOL     $R4,8               SHIFT TO LEFT HALF OF WORD
004293          11BE  9873                        LDR     $R1,+$B3            LOAD NEXT WORD
004294          11BF  9570 00FF                   AND     $R1,=X'FF'          STRIP TO 8 BITS
004295          11C1  C651                        XOR     $R4,=$R1            PUT WORD TOGETHER
004296          11C2  CF71                        STR     $R4,+$B1            STORE WORD IN THE PACKED LOCATION
004297          11C3  57F9                        BINC    $R5,>PKR1           BRANCH TO PACK ALL WORDS
004298          11C4  8FC0 0322                   RSTR    SAV1,=Z'CCD0'       RESTORE REGS.
                11C6  CCD0
004299          11C7  8384                        JMP     $B4
004300                                    *
004301                                    * WAIT FOR STATUS COMPLETE
004302                                    *   LNJ  $B4,<TEST
004303                                    *   B    $+2     RETURN
004304                                    *   DC   XX      TIMEOUT IN 120TH'S SEC.
004305                                    *
004306                                    *
004307          11C8  8F00 14E7           TEST    SAVE    <SAV1,=Z'4909'      R1,4,7,B4,7
                11CA  4909
004308          11CB  DB80 0000       X           LAB     $B5,<ZHPFR          CLEAR B5
004309          11CD  8751                        CL      =$R1
004310          11CE  9F00 0000       X           STR     $R1,<ZHRTC1         ZERO OUT RTC RESET VALUE
004311          11D0  1C02                        LDV     $R1,=2
004312          11D1  9F00 0000       X           STR     $R1,<ZHRTCL         SET FOR RUPT LEVEL 1
004313          11D3  9874                        LDR     $R1,+$B4            DUMMY TO INCREMENT B4
004314          11D4  9874                        LDR     $R1,+$B4            PICK UP TIMEOUT VALUE
004315          11D5  9F00 0000       X           STR     $R1,<ZHRTCC         SET REAL TIME CLOCK
004316          11D7  9F00 154B                   STR     $R1,<INTM
004317          11D9  0004                        RTCN
004318          11DA  C380 1161                   LNJ     $B4,<INXT           INPUT NEXT STATUS
004319          11DC  82D5                TESTZ   LB      =$R5,=Z'1000'       TEST FOR STATUS COMPLETE
                11DD  1000
004320          11DE  050F                        BBT     >TEST1              BRANCH IF COMPLETE
004321          11DF  8980 0000       X           CMZ     <ZHRTCC             TEST RTC
004322          11E1  0F01 FFFF                   NOP     $                   **********
004323                                    *
004324          11E3  C800 1561                   LDR     $R4,<CONT6          FUNCTION CODE
004325          11E5  C380 1173                   LNJ     $B4,<CGSCH
004326          11E7  8055                        IO      =$R5,=$R4           INPPT STATUS
                11E8  0054
004327          11E9  0773                        BIOT    >TESTZ              BRANCH MEANS TRY AGAIN
004328          11EA  F380 125C                   LNJ     $B7,<ERROR          INPUT STATUS WAS NAK'ED
004329          11EC  0000                        HLT
004330          11ED  0005                TEST1   RTCF                        TURN RTC OFF
004331          11EE  9800 154B                   LDR     $R1,<INTM           GET INITIAL TIME
004332          11F0  9200 0000       X           SUB     $R1,<ZHRTCC         GET ELAPSED
004333          11F2  C380 11F8                   LNJ     $B4,<UPTM           UPDATE TIME
004334                                    *
004335          11F4  8F80 14E7                   RSTR    <SAV1,=Z'4909'
                11F6  4909
004336          11F7  8384                        JMP     $B4                 EXIT
004337                                    *
004338                                    *
004339          11F8  9A00 154A           UPTM    ADD     $R1,<TTOT           ADD ELAPSED TO TAL
004340          11FA  9F00 154A                   STR     $R1,<TTOT
004341          11FC  1041                        SOR     $R1,1               DEVIDE TICKS BY 2
004342          11FD  9970 003C                   CMR     $R1,=60
004343          11FF  0205                        BL      >UPTM1
004344          1200  8700 154A                   CL      <TTOT               CLEAR TOTAL
004345          1202  8AB0 1549                   INC     <SEC                BUMP SECONDS COUNTER
004346          1204  8384              UPTM1   JMP     $B4
004347                                    *
004348                                    * TIME DELAY OF 25 MSEC (APPROX)
004349                                    *
004350                                    *     LNJ  $B4,<DLAY
004351                                    *
004352          1205  8F00 14E7           DLAY    SAVE    <SAV1,=Z'CE8C'
                1207  CE8C
```

```
004353  1208  9800 154E              LDR    $R1,<HRTZ              GET FREQ
004354  120A  1044                   SOR    $R1,4                  SET FOR CLOCK/16
004355  120B  0F90                   B      >DLAY2
004356                         *
004357                         * LONG DELAY (APPROX 225 MS)
004358                         *
004359                         *
004360  120C  8F40 02DA        DLAYLI SAVE   SAV1,=Z'CE8C'
        120E  CE8C
004361  120F  9800 154E              LDR    $R1,<HRTZ
004362  1211  1042                   SOR    $R1,2
004363  1212  1F03                   MLV    $R1,=3
004364  1213  0F81 0007              B      DLAY2
004365  1215  8F40 02D1        DLAYLG SAVE   SAV1,=Z'CE8C'
        1217  CE8C
004366  1218  9800 154E              LDR    $R1,<HRTZ              GET CLOCK
004367  121A  1041                   SOR    $R1,1                  GET CLOCK/4
004368  121B  DB80 0000        X     LAB    $B5,<ZHPFR             CLEAR B5
004369  121D  DF80 0001        X     STB    $B5,<ZH1SAZ+$AF        ZERO OUT LEV1 INTERRUPT VECTOR
004370  121F  9F00 0000        X     STR    $R1,<ZHRTCC            SET RTC CURRENT VALUE
004371  1221  C380 11F8              LNJ    $B4,<UPTM              UPDATE TIME
004372  1223  8751                   CL     =$R1
004373  1224  9F00 0000        X     STR    $R1,<ZHRTCI            RTC RESET VALUE
004374  1226  1C01                   LDV    $R1,=1
004375  1227  9F00 0000        X     STR    $R1,<ZHRTCL
004376  1229  C800 1561              LDR    $R4,<CONT6             INPUT STATUS FUNCTION
004377  122B  C380 1173              LNJ    $B4,<CGSCH             PUT IN CHAN NUMBER
004378                         *
004379  122D  1C0A                   LDV    $R1,=10
004380  122E  0004                   RTCN                          TURN ON RTC TIMER
004381  122F  1700 1236        DLAY1 BDEC   $R1,<DLAY4
004382  1231  1C0A                   LDV    $R1,=10                RESTORE COUNTER
004383  1232  4E06                   ADV    $R4,=6
004384                         *
004385                         * GIVE INPUT LCI BYTE TO TRY TO INDUCE INTERFERENCE
004386  1233  8055                   IO     =$R5,=$R4
        1234  0054
004387  1235  4EFA                   ADV    $R4,=-6                GET BACK INPUT STAT FC
004388  1236  E800 1581        DLAY4 LDR    $R6,<ERMG+1            GET TEST LABLE
004389  1238  E970 5349              CMR    $R6,=A'SI'
004390  123A  0985                   BNE    >DLAY3                 BRANCH MEANS NOT TEST SI
004391                         * IS SOFT INITIALIZE TEST. BYPASS STATUS INPUTS.
004392  123B  8AD1                   INC    =$R1
004393  123C  8980 0000        X     CMZ    <ZHRTCC
004394  123E  0F87                   B      >DLAY5
004395                         *
004396  123F  8980 0000        X     CMZ    <ZHRTCC
004397  1241  8055            DLAY3 IO     =$R5,=$R4              INPUT STAT
        1242  0054
004398                         * HANG HERE IF MLCC DOES NOT RESPOND WITH A SECOND HALF READ
004399  1243  DF00 154D              STR    $R5,<STAT              STORE LAST STATUS READ
004400  1245  0A00 122F        DLAY5 BAG    <DLAY1                 WAIT TILL DONE
004401  1247  0005                   RTCF                          TURN OFF TIMER
004402  1248  8FC0 029E              RSTR   SAV1,=Z'CE8C'
        124A  CE8C
004403  124B  8384                   JMP    $B4                    RETURN
004404                         *
004405                         *  SET UP RTC AND START IT.
004406                         *
004407                         *      LNJ  $B4,<RTC
004408                         *      DC   XX   XX = 120'S OF A SECOND DELAY DESIRED
004409                         *
004410  124C  9874            RTC    LDR    $R1,+$B4               GET DELAY DESIRED
004411  124D  9F00 0000        X     STR    $R1,<ZHRTCI            STORE INITIAL VALUE
004412  124F  9F00 0000        X     STR    $R1,<ZHRTCC            STORE CURRENT VALUE
004413  1251  CF80 14CD              STB    $B4,<TPR
004414  1253  C380 11F8              LNJ    $B4,<UPTM              UPDATE TIME
004415  1255  CC80 14CD              LDB    $B4,<TPR
004416  1257  1C03                   LDV    $R1,=3                 LEVEL 3
004417  1258  9F00 0000        X     STR    $R1,<ZHRTCL            FOR RTC
004418  125A  0004                   RTCN                          TURN ON RTC
004419  125B  8384                   JMP    $B4                    RETURN
004420                         *
004421                         *      EERROR PRINT ROUTINE
004422                         *
004423                         *EELNJ  $B7,<ERROR
004424                         *
004425  125C  8F40 02DA        ERROR  SAVE   SAV5,=Z'FFFF'          SAVE ALL REGISTERS
        125E  FFFF
004426  125F  1CF0                   LDV    $R1,=-16
004427  1260  8753                   CL     =$R3                   CLEAR INDEX
004428  1261  EBC7 FFFE              LAB    $B6,$B7,-2*$AF         DECREMENT $B7
004429  1263  EF80 126A              STB    $B6,<ERR1+4+$AF        STORE ERROR ADDRESS FOR ERROR CALL
004430                         ERR1   CALL   ZV$ER,$,ERMG
        1265  FBC0 0003
        1267  D380 0000        X
        1269  0F80
        126A  1265
        126B  1580
004431  126C  EB80 14C1              LAB    $B6,<CRLF              PUT CR-L INTO CALL
004432  126E  CB80 14C2              LAB    $B4,<SPACE
004433  1270  AB80 1282              LAB    $B2,<ERR3+4+2*$AF
004440  1272  EF82                   STB    $B6,$B2
004441  1273  8980 154C        ER11  CMZ    <ERF                   CHECK FLAG FOR SHORT REPORT
004442  1275  0900 128D              BE     <ERR5
004443  1277  BBB0 1537        ERR2  LAB    $B3,<SAV5,$R3          FECTH AND STORE REGISTER
004444  1279  9B80 1281              LAB    $B1,<ERR3+4+$AF
004450  127B  BF81            ERR21 STB    $B3,$B1
004458                         ERR3   CALL   ZV$THZ,$,$
        127C  FBC0 0003
        127E  D380 0000        X
        1280  0F80
        1281  127C
        1282  127C
004463  1283  8AD3            ERR34  INC    =$R3                   BUMP INDEX
004464  1284  CF82                   STB    $B4,$B2
004465  1285  3D06                   CMV    $R3,=7*$AF+1           SEE IF CR-LF NEEDED
004466  1286  0984                   BNE    >ERR4
004467  1287  EB80 14C1              LAB    $B6,<CRLF
004468  1289  EF82                   STB    $B6,$B2                PUT CR-LF INTO CALL
004469  128A  1780 1277        ERR4  BINC   $R1,<ERR2               DO NEXT REGISTER
004470  128C  0000                   HLT                           HALT AFTER ERROR
004471  128D  8F80 1537        ERR5  RSTR   <SAV5,=Z'FFFF'          RESTORE REGISTERS
```

```
                  168F  FFFF
004472  1290  8387          JMP     $B7                     RETURN
004483                SAFF  NULL
004484                *
004485                *
004486                *
004487                *       -VD-VL-VC-VC -VI-VN-VS-VT-VR-VU-VC-VT-VI-VO-VN -VT-VE-VS-VT , -VC-VH-
004488                *
004489                *
004490                *
004491                *
004492                *       RAM INSTRUCTION TEST PROGRAM GENERATION
004493                *
004494                *      THIS SECTION GETS LOADED AT HEX 200
004495                *
004496                *
004497                *        ORG     X'200'
004498          1291  INSTR1  EQU     $
004499                *        LOC     ST
004500          0200  ST      EQU     X'0200'
004501                *        B       BT1                     B BT1
004504  1291  2003
004505                *        WAIT    UNCONDITIONAL           BRANCH ERROR
004506  1292  BD31
004507                *        LOC     BT1
004508          0205  BT1     EQU     X'0205'
004509                *        LD      =1                      LD =1
004510  1293  00C6
004513                *        C       =1                      C =1
004514  1294  0186
004517  1295  0110
004518  1296  9741
004519                *        BET     BT6                     BRANCH IF TRUE
004520  1297  9641
004523  1298  2703
004524                *        LOC     BT5
004525          0210  BT5     EQU     X'0210'
004526                *        WAIT    BRANCH                  ON EQUAL TRUE OR 'LD' OR 'C' ERROR
004527  1299  BD31
004528                *        LOC     BT6
004529          0213  BT6     EQU     X'0213'
004530                *        BEF     BT5                     BRANCH IF FALSE
004531  129A  0096
004532  129B  4126
004535                *        LOC     RMOP
004536          0217  RMOP    EQU     X'0217'
004537                *        C       24                      COMPARE WITH LCT 24 (0)
004538  129C  0996
004541  129D  1810
004542  129E  9741
004543                *        BEF     BTC
004544  129F  9641
004547  12A0  2603
004548                *        LOC     BTB
004549          0220  BTB     EQU     X'0220'
004550                *        WAIT    BRANCH                  ON EQUAL FALSE ERROR
004551  12A1  BD31
004552                *        LOC     BTC
004553          0223  BTC     EQU     X'0223'
004554                *        BET     BT5
004555  12A2  0096
004556  12A3  4127
004559                *        DEC     R                       GOES FROM 1 TO 0
004560  12A4  E95A
004561                *        BZT     BT12                    BRANCH IF ZERO TRUE
004562  12A5  5D27
004565                *        LOC     BT11
004566          022B  BT11    EQU     X'022B'
004567                *        WAIT    BRANCH                  ON ZERO TRUE ERROR
004568  12A6  03BD
004569  12A7  3100
004570                *        LOC     BT12
004571          022E  BT12    EQU     X'022E'
004572                *        BZF     BT17                    BRANCH IF ZERO FALSE
004573  12A8  5D26
004576                *        DEC     R                       GOES FROM 0 TO FF
004577  12A9  045A
004578                *        BZF     BT18
004579  12AA  5D26
004582                *        LOC     BT17
004583          0235  BT17    EQU     X'0235'
004584                *        WAIT    BRANCH                  ON ZERO FALSE ERROR
004585  12AB  03BD
004586  12AC  3100
004587                *        LOC     BT18
004588          0238  BT18    EQU     X'0238'
004589                *        BLCT    ER21
004590  12AD  DE63
004593  12AE  2704
004594                *        BLCF    GN22
004595  12AF  DE63
004598  12B0  2603
004599                *        LOC     ER21
004600          0240  ER21    EQU     X'0240'
004601                *        WAIT    BRANCH
004602  12B1  BD31
004603                *        LOC     GN22
004604          0243  GN22    EQU     X'0243'
004605                *        GNB
004606  12B2  00BD
004607  12B3  3120
004608                *        BLCF    ER21
004609  12B4  DE63
004612  12B5  26F6
004613                *        LOC     BT1D
004614          024A  BT1D    EQU     X'024A'
004615                *        C       23                      COMPARE WITH LCT 23
004618  12B6  9617
004619  12B7  1097
004620                *        BET     BT4F
004621  12B8  4196
004622  12B9  4127
004625                *        WAIT    ERROR                   IN STORE INST.
```

```
004626  12BA  03BD
004627  12BB  3100
004628                *         LOC     BT4F
004629        0256    BT4F      EQU     X'0256'
004630                *         AND     23          AND R (FF) WITH LCT 23 (FF)
004633  12BC  D417
004634                *         C       =X'FF'      C =X'FF'
004637  12BD  86FF
004638  12BE  1097
004639                *         BEF     BT4D
004640  12BF  4196
004641  12C0  4126
004644                *         AND     24          AND R (FF) WITH LCT 24 (0)
004645  12C1  16D4
004646                *         C       24          COMPARE R(0) WITH LCT 24 (0)
004649  12C2  1896
004652  12C3  1810
004653  12C4  9741
004654                *         BEF     BT4D
004655  12C5  9641
004658  12C6  260B
004659                *         AND     23          AND R (0) WITH LCT 23 (FF)
004662  12C7  D417
004663                *         C       24          COMPARE R (0), LCT 24 (0)
004666  12C8  9618
004667  12C9  1097
004668                *         BET     BTAC
004669  12CA  4196
004670  12CB  4127
004673                *         LOC     BT4D
004674        0277    BT4D      EQU     X'0277'
004675                *         WAIT    AND         INSTRUCTION ERROR
004676  12CC  03BD
004677  12CD  3100
004678                *         LOC     BTAC
004679        027A    BTAC      EQU     X'027A'
004680                *         AND     24          AND R (0) WITH LCT 24 (0)
004683  12CE  D418
004684                *         B       TMP1
004687  12CF  2002
004688                *         LOC     TEMPA
004689        027E    TEMPA     EQU     X'027E'
004690                *         B       ST          BRABCH TO SUBROUTINE
004693  12D0  2080
004694                *         LOC     TMP1
004695        0280    TMP1      EQU     X'0280'
004696                *         C       24
004699  12D1  9618
004700  12D2  1097
004701                *         BEF     BT4D
004702  12D3  4196
004703  12D4  4126
004706                *         OR      24          OR R (0), LCT 24 (0)
004707  12D5  EEDA
004710                *         C       24          COMPARE R WITH DATA BUFF
004711  12D6  1896
004714  12D7  1810
004715  12D8  9741
004716                *         BEF     BT4C
004717  12D9  9641
004720  12DA  2616
004721                *         OR      23          OR R(0), LCT 23 (FF)
004724  12DB  DA17
004725                *         C       23          COMPARE R WITH LCT 23
004728  12DC  9617
004729  12DD  1097
004730                *         BEF     BT4C
004731  12DE  4196
004732  12DF  4126
004735                *         OR      24          OR R (FF), LCT 24 (0)
004736  12E0  0BDA
004739                *         C       23          COMPARE R (FF), LCT 23 (FF)
004740  12E1  1896
004743  12E2  1710
004744  12E3  9741
004745                *         BET     BTRC
004746  12E4  9641
004749  12E5  2703
004750                *         LOC     BT4C
004751        02AA    BT4C      EQU     X'02AA'
004752                *         WAIT    OR          INSTRUCTION ERROR
004753  12E6  BD31
004754                *         LOC     BTRC
004755        02AD    BTRC      EQU     X'02AD'
004756                *         OR      23          OR R (FF), LCT23 (FF)
004757  12E7  00DA
004760                *         C       23          COMPARE R (FF), LCT 23
004761  12E8  1796
004764  12E9  1710
004765  12EA  9741
004766                *         BEF     BT4C
004767  12EB  9641
004770  12EC  26F2
004771                *         XOR     24          EXCL. OR R (FF), DATA BUFF  (0)
004774  12ED  D818
004775                *         C       23          COMPARE R(FF), LCT 23 (FF)
004778  12EE  9617
004779  12EF  1097
004780                *         BEF     BT4B
004781  12F0  4196
004782  12F1  4126
004785                *         XOR     23          EXCLUSIVE OR R (FF), LCT 23 (FF)
004786  12F2  16D8
004789                *         C       24          COMPARE R (0) WITH LCT 24 (0)
004790  12E3  1796
004793  12F4  1810
004794  12F5  9741
004795                *         BEF     BT4B
004796  12F6  9641
004799  12F7  260B
004800                *         XOR     24          EXCL. OR R (0) WITH LCT 24 (0)
004803  12F8  D818
004804                *         C       24
```

```
004807  12F9  9618
004808  12FA  1097
004809                *       BET     BTXC
004810  12FB  4196
004811  12FC  4127
004814                *       LOC     BT4B
004815        02D9    BT4B    EQU     X'02D9'
004816                *       WAIT    XOR                     INSTRUCTION ERROR
004817  12FD  03BD
004818  12FE  3100
004819                *       LOC     BTXC
004820        02DC    BTXC    EQU     X'02DC'
004821                *       XOR     23                      EXCL. OR R (0), LCT 23 (FF)
004824  12FF  0B17
004825                *       C       23                      COMPARE R (FF), LCT 23  (FF)
004826  1300  9617
004829  1301  1097
004830                *       BEF     BT4B
004831  1302  4196
004832  1303  4126
004835                *       LOC     BT4J
004836        02E7    BT4J    EQU     X'02E7'
004837                *       LD      =X'AA'                   LD =X'AA'
004838  1304  F2C6
004841                *       RET
004842  1305  AADE
004843  1306  526E
004844                *       NOP
004845  1307  0001
004846                *
004847                * RAM INSTRUCTION TEST, PART 2, LOADED AT 27A
004848                *
004849                *
004850        1308    INSTR2  EQU     $
004851                *       LOC     INS2A
004852        02EE    INS2A   EQU     X'02EE'
004853                *       JUMP    INS6
004856  1308  3F00
004857                *       ORG     X'02F1'
004858                *       LOC     BTYD
004859        02F1    BTYD    EQU     X'02F1'
004860                *       NOP     DUMMY
004861  1309  EF01
004862                *       BS      TEMPA                   BRANCH STORE TO X'200'
004863  130A  BD30
004864  130B  8020
004867                *       C       =X'AA'                  COMPARE R(AA), =AA
004868  130C  8786
004871  130D  AA10
004872  130E  9741
004873                *       BET     BTA7
004874  130F  9641
004877  1310  2703
004878                *       LOC     BTB7
004879        0300    BTB7    EQU     X'0300'
004880                *       WAIT    BRANCH                  AND STORE ERROR
004881  1311  BD31
004882                *       LOC     BTA7
004883        0303    BTA7    EQU     X'0303'
004884                *       BS      BTA6                    BRANCH STORE TO X'306'
004885  1312  00BD
004886  1313  3080
004889  1314  2062
004890                *       C       =X'55'                   C =X'55'
004893  1315  8655
004894  1316  1097
004895                *       BEF     BTB7
004896  1317  4196
004897  1318  4126
004900                *
004901                * HERE FOLLOWS THE TABLE LOOKUP TEST
004902                *
004903                *       LD      =0                      LD =0
004904  1319  EFC6
004907                *       TLU     55                      TABLE LOOKUP, POINTER AT LCT 55
004908  131A  00DE
004911  131B  37BD
004912  131C  30E0
004913                *       C       =3                      FIRST ENTRY OF TABLE IS 3
004916  131D  8603
004917  131E  1097
004918                *       BET     NEXT
004919  131F  4196
004920  1320  4127
004923                *       WAIT    TABLE                   LOOKUP ERROR 24 1ST ENTRY
004924  1321  03BD
004925  1322  3100
004926                *       LOC     NEXT
004927        0324    NEXT    EQU     X'0324'
004928                *       LD      =2                      LD =2, GET 3D ENTRY
004931  1323  C602
004932                *       TLU     55                      TABLE LOOKUP, LCT 55
004935  1324  DE37
004936  1325  BD30
004937                *       WAIT
004938  1326  E0BD
004939  1327  3100
004940                *       WAIT
004941  1328  BD31
004942                *       WAIT
004943  1329  00BD
004944  132A  3100
004945                *       WAIT    TLU                     ERROR 24 INCORRECT BRANCH
004946  132B  BD31
004947                *       B       NEXT1                   GOOD
004948  132C  0020
004951                *       WAIT    BAD                     TLU BRANCH
004952  132D  09BD
004953  132E  3100
004954                *       WAIT    DITTO
004955  132F  BD31
004956                *       WAIT    DITTO
004957  1330  00BD
```

```
004958  1331  3100
004959                        *       LOC     NEXT1
004960        0342            NEXT1   EQU     X'0342'
004961                        *       C       =X'2'           COMPARE R WITH =X'2'
004964  1332  8602
004965  1333  1097
004966                        *       BET     NEXT2
004967  1334  4196
004968  1335  4127
004971                        *       WAIT    TLU             ERROR 24 WRONG DATA CAME BACK
004972  1336  03BD
004973  1337  3100
004974                        *       LOC     NEXT2
004975        034E            NEXT2   EQU     X'034E'
004976                        *       LD      =1              GET SECOND ENTRY FROM TABLE
004979  1338  C601
004980                        *       TLU     55              TLU INDIRECT THRU LCT 55
004983  1339  DE37
004984  133A  BD30
004985                        *       C       =X'1A'
004986  133B  E086
004989  133C  1A10
004990  133D  9741
004991                        *       BET     NEXT3
004992  133E  9641
004995  133F  2703
004996                        *       WAIT    WRONG           DATA BACK FROM TLU
004997  1340  BD31
004998                        *       LOC     NEXT3
004999        0361            NEXT3   EQU     X'0361'
005000                        *       WAIT
005001  1341  00BD
005002  1342  3100
005003                        *       NOP
005004                        *
005005                        * INSTRUCTION TEST, PART 3, GETS LOADED AT 306
005006                        *
005007                        *
005008        1343            INSTR3  EQU     $
005009                        *       ORG     X'036A'
005010  1343  0100
005011                        *       LOC     BTA6
005012        036A            BTA6    EQU     X'036A'
005013                        *       LD      3               LD  R FROM LCT 3 (FF), PRELOADED IN LCT SETU
005016  1344  D603
005017                        *       C       =X'FF'           C ='FF'
005020  1345  86FF
005021  1346  1097
005022                        *       BEF     BTA9
005023  1347  4196
005024  1348  4126
005027                        *       LD      28              LD R, LCT 28 (AA), PRELOADED
005028  1349  0BD6
005031                        *       C       =X'AA'
005032  134A  1C86
005035  134B  AA10
005036  134C  9741
005037                        *       BET     BTB5
005038  134D  9641
005041  134E  2703
005042                        *       LOC     BTA9
005043        0380            BTA9    EQU     X'0380'
005044                        *       WAIT    LOAD            ADDRESSING ERROR
005045  134F  BD31
005046                        *       LOC     BTB5
005047        0383            BTB5    EQU     X'0383'
005048                        *       LD      44              LOAD R,LCT 44 (55), PRE LOADED
005049  1350  00D6
005052                        *       C       =X'55'           =X'55'
005053  1351  2C86
005056  1352  5510
005057  1353  9741
005058                        *       BEF     BTA9
005059  1354  9641
005062  1355  26F2
005063                        *       LD      63              LOAD  R, LCT 63 (AA), PRELOADED
005066  1356  D63F
005067                        *       C       =X'AA'           C =X'AA'
005070  1357  86AA
005071  1358  1097
005072                        *       BEF     BTA9
005073  1359  4196
005074  135A  4126
005077                        *       LD      =X'80'          TEST VALUE
005078  135B  E7C6
005081                        *       SR      SHIFT           RIGHT
005082  135C  8054
005083                        *       C       =X'40'
005086  135D  8640
005087  135E  1097
005088                        *       BEF     ERR6            SHIFT FAILED IF BRANCH
005089  135F  4196
005090  1360  4126
005093                        *       SR      SHIFT           AGAIN
005094  1361  1F54
005095                        *       C       =X'20'
005098  1362  8620
005099  1363  1097
005100                        *       BEF     ERR6            BRANCH IF SHIFT FAILED
005101  1364  4196
005102  1365  4126
005105                        *       BLBF    BT4G            BRANCH IF LAST BLOCK FALSE
005106  1366  1596
005107  1367  6585
005108  1368  2027
005111                        *       LOC     BT4H
005112        03B5            BT4H    EQU     X'03B5'
005113                        *       WAIT    BLCF            OR BLCT FAILURE
005114  1369  03BD
005115  136A  3100
005116                        *       LOC     BT4G
005117        03B8            BT4G    EQU     X'03B8'
005118                        *       BLBT    BT4H            BRANCH IF LAST BLOCK TRUE
```

```
005119  136B  9665
005120  136C  8520
005123  136D  26F7
005124                    *       LD      =X'55'                  LD =X'55'
005127  136E  C655
005128                    *       RET
005129  136F  DE52
005130  1370  6E00
005131                    *       LOC     ERR6
005132        03C4        ERR6    EQU     X'03C4'
005133                    *       WAIT
005134  1371  BD31
005135                    *       NOP
005136  1372  0001
005137                    *
005138                    *
005139                    * INSTRUCTION TEST - PART 4 - LOADED AT X'5FD'
005140                    *
005141        1373        INSTR4  EQU     $
005142                    *       WAIT
005143  1373  BD31
005144                    *       WAIT
005145  1374  00BD
005146  1375  3100
005147                    *       LOC     INS4
005148        03CE        INS4    EQU     X'03CE'
005149                    *       JUMP    INS5
005152  1376  3F00
005153                    *
005154                    * PART 5 - LOADED AT AFE
005155                    *
005156        1377        INSTR5  EQU     $
005157                    *       WAIT
005158  1377  06BD
005159  1378  3100
005160                    *       WAIT
005161  1379  BD31
005162                    *       LOC     INS5
005163        03D7        INS5    EQU     X'03D7'
005164                    *       JUMP    BTYD
005165  137A  003F
005168  137B  FF17                DC      BTYD-X'03DA'
005169                    *
005170                    * INSTRUCTION TEST - PART 6, LOADED AT DFB
005171                    *
005172        137C        INSTR6  EQU     $
005173                    *       WAIT
005174  137C  BD31
005175                    *       WAIT
005176  137D  00BD
005177  137E  3100
005178                    *       LOC     INS6
005179        03E0        INS6    EQU     X'03E0'
005180                    *       JUMP    INS4
005183  137F  3FFF
005184
005185                    * TRANSMIT DATA CHANNEL PROGRAM
005186                    *               GETS LOADED AT X'400' IN MLCC RAM
005187                    *
005188                    *
005189                    *
005190                    *       ORG     X'400'
005191  1380  EB00
005192                    *       LOC     XMDT
005193        0400        XMDT    EQU     X'0400'
005194        1381        TRANM1  EQU     $
005195                    *       B       RTD1
005198  1381  2003
005199                    *       LOC     RTD
005200        0402        RTD     EQU     X'0402'
005201                    *       WAIT    NORMAL                  END
005202  1382  BD31
005203                    *       LOC     RTD1
005204        0405        RTD1    EQU     X'0405'
005205                    *       LD      55                      LOAD AND CHECK COUNTER
005206  1383  00D6
005209                    *       C       =5                      IF 5 STORE IN 57
005210  1384  3786
005213  1385  0510
005214  1386  9741
005215                    *       BET     R57
005216  1387  9641
005219  1388  275A
005220                    *       C       =4                      IF 4 STORE IN 58
005223  1389  8604
005224  138A  1097
005225                    *       BET     R58
005226  138B  4196
005227  138C  4127
005230                    *       C       =3                      IF 3 STORE IN 59
005231  138D  5886
005234  138E  0310
005235  138F  9741
005236                    *       BET     R59
005237  1390  9641
005240  1391  2756
005241                    *       C       =2                      IF 2 STORE IN 60
005244  1392  8602
005245  1393  1097
005246                    *       BET     R60
005247  1394  4196
005248  1395  4127
005251                    *       C       =1                      IF 1 STORE IN 61
005252  1396  5486
005255  1397  0110
005256  1398  9741
005257                    *       BET     R61
005258  1399  9641
005261  139A  2752
005262                    *       LD,     IS                      ZERO
005263  139B  BD30
005264                    *       ST      62                      STORE IN 62
005265  139C  A0D7
```

```
005268                          *       LOC     RTD2
005269          0439            RTD2    EQU     X'0439'
005270                          *       LD      56              CHECK FLAG
005271  1390  3ED6
005274                          *       C       =0
005275  1391  3886
005278  1392  0010
005279  1393  9741
005280                          *       BET     R70             BRANCH IF EOR
005281  1394  9641
005284  1395  2710
005285                          *       BLCF    R67
005286  1396  DE63
005289  1397  2603
005290                          *       WAIT    GNB             BLCF DID NOT BRANCH
005291  1398  BD31
005292                          *       LOC     R67
005293          044B            R67     EQU     X'044B'
005294                          *       BLCT    R68
005295  1399  00DE
005296  139A  6327
005299                          *       GNB
005300  139B  15BD
005301  139C  3120
005302                          *       B       RTD
005305  139D  20AE
005306                          *
005307                          *
005308                          *
005309                          *       LOC     R70
005310          0454            R70     EQU     X'0454'
005311                          *       BLCF    R69
005312  139E  DE63
005315  139F  260F
005316                          *       BLCT    R71
005317  13A0  DE63
005320  13A1  2703
005321                          *       WAIT    WAIT            BLCF DID NOT BRANCH
005322  13A2  BD31
005323                          *       LOC     R71
005324          045F            R71     EQU     X'045F'
005325                          *       GNB
005326  13A3  00BD
005327  13A4  3120
005328                          *       B       RTD
005331  13A5  209E
005332                          *
005333                          *
005334                          *
005335                          *       LOC     R68
005336          0464            R68     EQU     X'0464'
005337                          *       WAIT    BLCT            DID BRANCH
005338  13A6  BD31
005339                          *       LOC     R69
005340          0467            R69     EQU     X'0467'
005341                          *       WAIT    BLCF            DID BRANCH
005342  13A7  00BD
005343  13A8  3100
005344                          *
005345                          *
005346                          *
005347                          *       LOC     R57
005348          046A            R57     EQU     X'046A'
005349                          *       LD
005350  13A9  BD30
005351                          *       ST      57
005352  13AA  A0D7
005355                          *       B       R56
005356  13AB  3920
005359                          *       LOC     R58
005360          0471            R58     EQU     X'0471'
005361                          *       LD
005362  13AC  1ABD
005363  13AD  30A0
005364                          *       ST      58
005367  13AE  D73A
005368                          *       B       R56
005371  13AF  2013
005372                          *       LOC     R59
005373          0478            R59     EQU     X'0478'
005374                          *       LD
005375  13B0  BD30
005376                          *       ST      59
005377  13B1  A0D7
005380                          *       B       R56
005381  13B2  3B20
005384                          *       LOC     R60
005385          047F            R60     EQU     X'047F'
005386                          *       LD
005387  13B3  0CBD
005388  13B4  30A0
005389                          *       ST      60
005392  13B5  D73C
005393                          *       B       R56
005396  13B6  2005
005397                          *       LOC     R61
005398          0486            R61     EQU     X'0486'
005399                          *       LD
        13B7  BD30
                                *       ST      61
        13B8  A0D7
                                *       LOC     R56
              048B              R56     EQU     X'048B'
                                *       LD      55
        13B9  3DD6
                                *       DEC
        13BA  375A
                                *       ST      55
        13BB  D737
                                *       LD      63
        13BC  D63F
                                *       DEC
                                *       ST      63
```

```
005423  13CA  5AD7
005426                  *       BZT     RTD2
005427  13CB  3F5D
005430  13CC  27A1
005431                  *       BLCT    R55
005432  13CD  DE63
005435  13CE  2703
005436                  *       JUMP    RTD1
005439  13CF  3FFF
005440                  * COME HERE FOR UNDERRUN TEST ONLY
005441                  *       LOC     R55
005442          049F    R55     EQU     X'049F'
005443                  *       GNB
005444  13D0  66BD
005445  13D1  3120
005446                  *       LD
005447  13D2  BD30
005448                  *       JUMP    RTD
005449  13D3  A03F
005452  13D4  FF5A              DC      RTD-X'04A8'
005453                  *       NOP
005454                  *       NOP
005455  13D5  0101
005456                  *       NOP
005457                  *
005458                  *
005459                  *
005460                  * RECEIVE DATA CHANNEL PROGRAM
005461                  *
005462                  *
005463                  *
005464                  *       ORG     X'200'
005465  13D6  0100      RCDTC1  EQU     $
005466          13D7    *       LOC     RCDT
005467                  RCDT    EQU     X'0200'
005468          0200    *       B       XMD1
005469
005472  13D7  2003
005473                  *       LOC     XMD
005474          0202    XMD     EQU     X'0202'
005475                  *       WAIT    NORMAL                  END
005476  13D8  BD31
005477                  *       LOC     XMD1
005478          0205    XMD1    EQU     X'0205'
005479                  *       LD      55                      LOAD AND CHECK COUNTER
005480  13D9  00D6
005483                  *       C       =5                      IF 5 LOAD FROM 57
005484  13DA  3786
005487  13DB  0510
005488  13DC  9741
005489                  *       BET     XM57
005490  13DD  9641
005493  13DE  2755
005494                  *       C       =4                      IF 4 LOAD FROM 58
005497  13DF  8604
005498  13E0  1097
005499                  *       BET     XM58
005500  13E1  4196
005501  13E2  4127
005504                  *       C       =3                      IF 3 LOAD FROM 59
005505  13E3  5386
005508  13E4  0310
005509  13E5  9741
005510                  *       BET     XM59
005511  13E6  9641
005514  13E7  2751
005515                  *       C       =2                      IF 2 LOAD FROM 60
005518  13E8  8602
005519  13E9  1097
005520                  *       BET     XM60
005521  13EA  4196
005522  13EB  4127
005525                  *       C       =1                      IF 1 LOAD FROM 61
005526  13EC  4F86
005529  13ED  0110
005530  13EE  9741
005531                  *       BET     XM61
005532  13EF  9641
005535  13F0  274D
005536                  *       LD      62                      IS ZERO
005539  13F1  D63E
005540                  *       ST,     PUT                     IN DATA BUFFER
005541  13F2  BD30
005542                  *       LOC     XMD2
005543          0239    XMD2    EQU     X'0239'
005544                  *       LD      56                      CHECK FLAG
005545  13F3  C0D6
005548                  *       C       =0
005549  13F4  3886
005552  13F5  0010
005553  13F6  9741
005554                  *       BET     XM70
005555  13F7  9641
005558  13F8  2710
005559                  *       BLCF    XM67
005560  13F9  DE63
005563  13FA  2603
005564                  *       WAIT    GNB                     BLCF DID NOT BRANCH
005565  13FB  BD31
005566                  *       LOC     XM67
005567          024B    XM67    EQU     X'024B'
005568                  *       BLCT    XM68
005569  13FC  00DE
005570  13FD  6327
005573                  *       LOC     XM71
005574          024F    XM71    EQU     X'024F'
005575                  *       GNB
005576  13FE  10BD
005577  13FF  3120
005578                  *       B       XMD
005581  1400  20AE
005582                  *
005583                  *
```

```
005584                          *
005585                          *       LOC     XM70
005586          0254            XM70    EQU     X'0254'
005587                          *       BLCF    XM69
005588  1401    DE63
005591  1402    260A
005592                          *       BLCT    XM71
005593  1403    DE63
005596  1404    27F3
005597                          *       WAIT    WAIT                    BLCT DID NOT BRANCH
005598  1405    BD31
005599                          *
005600                          *
005601                          *
005602                          *       LOC     XM68
005603          025F            XM68    EQU     X'025F'
005604                          *       WAIT    BLCT                    DID BRANCH
005605  1406    00BD
005606  1407    3100
005607                          *       LOC     XM69
005608          0262            XM69    EQU     X'0262'
005609                          *       WAIT    BLCF                    DID BRANCH
005610  1408    BD31
005611                          *
005612                          *
005613                          *
005614                          *       LOC     XM57
005615          0265            XM57    EQU     X'0265'
005616                          *       LD      57
005617  1409    00D6
005620                          *       ST
005621  140A    39BD
005622  140B    30C0
005623                          *       B       XM56
005626  140C    201A
005627                          *       LOC     XM58
005628          026C            XM58    EQU     X'026C'
005629                          *       LD      58
005632  140D    D63A
005633                          *       ST
005634  140E    BD30
005635                          *       B       XM56
005636  140F    C020
005639                          *       LOC     XM59
005640          0273            XM59    EQU     X'0273'
005641                          *       LD      59
005642  1410    13D6
005645                          *       ST
005646  1411    3BBD
005647  1412    30C0
005648                          *       B       XM56
005651  1413    200C
005652                          *       LOC     XM60
005653          027A            XM60    EQU     X'027A'
005654                          *       LD      60
005657  1414    D63C
005658                          *       ST
005659  1415    BD30
005660                          *       B       XM56
005661  1416    C020
005664                          *       LOC     XM61
005665          0281            XM61    EQU     X'0281'
005666                          *       LD      61
005667  1417    05D6
005670                          *       ST
005671  1418    3DBD
005672  1419    30C0
005673                          *       LOC     XM56
005674          0286            XM56    EQU     X'0286'
005675                          *       LD      55
005678  141A    D637
005679                          *       DEC
005680                          *       ST      55
005681  141B    5AD7
005684                          *       LD      63
005685  141C    37D6
005688                          *       DEC
005689  141D    3F5A
005690                          *       ST      63
005693  141E    D73F
005694                          *       BZT     XMD2
005695  141F    5D27
005698                          *       BLCT    XM55
005699  1420    A6DE
005700  1421    6327
005703                          *       JUMP    XMD1
005704  1422    033F
005707  1423    FF6B                    DC      XMD1-X'029A'
005708                          *       LOC     XM55
005709          029A            XM55    EQU     X'029A'
005710                          *       GNB                             FOR CCB OVERRUN TEST ONLY
005711  1424    BD31
005712                          *       ST
005713  1425    20BD
005714  1426    30C0
005715                          *       JUMP    XMD
005718  1427    3FFF
005719                          *       NOP
005720  1428    5F01
005721                          *       NOP
005722                          *       NOP
005723  1429    0101
005724                          *
005725                          *
005726                          *
005727                          *
005728                          *       MLCC INPUT DATA PROGRAM WITH
005729                          *         CRC TO GENERATE RANDOM DATA
005730                          *
005731                          *
005732                          *
005733                          *       ORG     X'200'
005734          142A            RCRT1   EQU     $
```

```
005735                          *       LOC     RMCRT
005736          0200            RMCRT   EQU     X'0200'
005737                          *       LD      2               CHECK CONF. IS LRC8
005740  142A  D602
005741                          *       C       =X'C6'
005744  142B  86C6
005745  142C  1097
005746                          *       BEF     RMCRT3          BRANCH IF NOT LRC8
005747  142D  4196
005748  142E  4126
005751                          *       LD      =1
005752  142F  15C6
005755                          *       LOC     RMCRT5
005756          020D            RMCRT5  EQU     X'020D'
005757                          *       ST      24
005758  1430  01D7
005761                          *       CCH
005762  1431  18BD
005763  1432  3160
005764                          *       BS      RMCRT4
005765  1433  BD30
005766  1434  8020
005769                          *       LD      24
005770  1435  30D6
005773                          *       XOR     =X'FF'
005774  1436  18C8
005777                          *       DEC
005778  1437  FF5A
005779                          *       XOR     =X'FF'
005782  1438  C8FF
005783                          *       B       RMCRT5
005786  1439  20ED
005787                          *       LOC     RMCRT3
005788          0220            RMCRT3  EQU     X'0220'
005789                          *       C       =X'44'          SEE IF CRC12
005792  143A  8644
005793  143B  1097
005794                          *       LD      =5
005795  143C  41C6
005798                          *       BET     RMCRT7
005799  143D  0596
005800  143E  4127
005803                          *       LOC     RMCRT1
005804          022B            RMCRT1  EQU     X'022B'
005805                          *       CCH
005806  143F  0DBB
005807  1440  3160
005808                          *       BS      RMCRT4
005809  1441  BD30
005810  1442  8020
005813                          *       LD      4
005814  1443  14D6
005817                          *       DEC
005818  1444  045A
005819                          *       B       RMCRT1
005822  1445  20F3
005823                          *
005824                          *       LOC     RMCRT7
005825          0238            RMCRT7  EQU     X'0238'
005826                          *       CCH
005827  1446  BD31
005828                          *       BS      RMCRT4
005829  1447  60BD
005830  1448  3080
005833  1449  2007
005834                          *       LD      4
005837  144A  D604
005838                          *       DEC
005839                          *       AND     =X'3F'
005840  144B  5AC4
005843                          *       B       RMCRT7
005844  144C  3F20
005847                          *       LOC     RMCRT4
005848          0247            RMCRT4  EQU     X'0247'
005849                          *       LD      4
005850  144D  F1D6
005853                          *       ST
005854  144E  04BD
005855  144F  30C0
005856                          *       LD      3
005859  1450  D603
005860                          *       ST
005861  1451  BD30
005862                          *       BLCF    RMCRT8
005863  1452  C0DE
005864  1453  6326
005867                          *       GNB
005868  1454  03BD
005869  1455  3120
005870                          *       LOC     RMCRT8
005871          0258            RMCRT8  EQU     X'0258'
005872                          *       LD      23
005875  1456  D617
005876                          *       DEC
005877                          *       ST      23
005878  1457  5AD7
005881                          *       BZT     RMCRT2
005882  1458  175D
005885  1459  2704
005886                          *       RET
005887  145A  DE52
005888  145B  6E00
005889                          *       LOC     RMCRT2
005890          0264            RMCRT2  EQU     X'0264'
005891                          *       WAIT
005892  145C  BD31
005893                          *       B       RMCRT
005894  145D  0020
005897                          *       NOP
005898  145E  9701
005899                          *
005900                          *
005901                          *       MLCC OUTPUT DATA PROGRAM
```

```
005902                          *
005903                          *
005904                          *
005905                          *       ORG     X'200'
005906          145F            ROD1    EQU     $
005907                          *       LOC     RMOPT
005908          0200            RMOPT   EQU     X'0200'
005909                          *       LD      =4
005912  145F    C604            *       ST      23
005913
005916  1460    D717            *       LOC     RMOPTA
005917          0204            RMOPTA  EQU     X'0204'
005919                          *       LD      =125
005922  1461    C67D            *       ST      24
005923
005926  1462    D718            *       LOC     RMOPT1
005927          0208            RMOPT1  EQU     X'0208'
005928                          *       LD
005929
005930  1463    BD30            *       CCH
005931
005932  1464    A0BD
005933  1465    3160
005934                          *       LD
005935  1466    BD30            *       BLCF    RMOPT2                  CHECK IF LAST CHARACTER
005936
005937  1467    A0DE
005938  1468    6326
005941                          *       GNB
005942  1469    03BD
005943  146A    3120
005944                          *       LOC     RMOPT2
005945          0218            RMOPT2  EQU     X'0218'
005946                          *       CCH
005947  146B    BD31            *       LD      24
005948
005949  146C    60D6            *       DEC
005952
005953  146D    185A            *       C       =0
005954
005957  146E    8600
005958  146F    1097
005959                          *       ST      24
005960  1470    41D7            *       BEF     RMOPT1
005963
005964  1471    1896
005965  1472    4126
005968                          *       LD      23
005969  1473    DFD6            *       DEC
005972
005973  1474    175A            *       C       =0
005974
005977  1475    8600
005978  1476    1097
005979                          *       ST      23
005980  1477    41D7            *       BEF     RMOPTA
005983
005984  1478    1796
005985  1479    4126
005988                          *       WAIT
005989  147A    CDBD
005990  147B    3100
005991                          *       NOP
005992                          *
005993                          *
005994                          *       NOP
005995  147C    0101
005996                          *
005997                          * BVT, BVF TEST
005998          147D            VALTST  EQU     $
005999                          *       ORG     X'200'
006000                          *       LD      =4                      COUNTER FOR 4 CCB'S
006003  147D    C604
006004                          *       LOC     BVT
006005          0202            BVT     EQU     X'0202'
006006                          *       BVBT    BVFF
006007  147E    9665
006008  147F    8540
006011  1480    2606
006012                          *       WAIT    ERROR
006013  1481    BD31            *       WAIT    ERROR, BVT              SHOULD BRANCH
006014
006015  1482    00BD
006016  1483    3100
006017                          *       LOC     BVFF
006018          020E            BVFF    EQU     X'020E'
006019                          *       BVBF    BVTT
006020  1484    9665
006021  1485    8540
006024  1486    2702
006025                          *       B       BWAIT
006028  1487    2006
006029                          *       LOC     BVTT
006030          0216            BVTT    EQU     X'0216'
006031                          *       WAIT
006032  1488    BD31            *       WAIT    ERROR, BVF              SHOULDN'T BRANCH
006033
006034  1489    00BD
006035  148A    3100
006036                          *       LOC     BWAIT
006037          021C            BWAIT   EQU     X'021C'
006038                          *       GNB
006039  148B    BD31            *       DEC
006040
006041  148C    205A            *       BZF     BVT                     BRANCH IF MORE CCB'S
006042
006043  148D    5D26            *       WAIT    WAIT                    FOR NEXT START IO
006046
006047  148E    DFBD
006048  148F    3100
006049                          *  CCB LIST RESET HAS NOW RESET VALID BITS
006050                          *       LD      =4                      COUNTER
```

```
006053  1490  C604
006054                        *       LOC     BVF
006055        0228            BVF     EQU     X'0228'
006056                        *       BVBF    BVR
006057  1491  9665
006058  1492  8540
006059  1493  2706
006062                        *       WAIT
006063  1494  BD31
006064                        *       WAIT
006065  1495  00BD
006066  1496  3100
006067                        *       LOC     BVR
006068        0234            BVR     EQU     X'0234'
006069                        *       BVBT    BVR
006070  1497  9665
006071  1498  8540
006074  1499  2602
006075                        *       B       BVL
006078  149A  2006
006079                        *       LOC     BVR
006080        023C            BVR     EQU     X'023C'
006081                        *       WAIT
006082  149B  BD31
006083                        *       WAIT
006084  149C  00BD
006085  149D  3100
006086                        *       LOC     BVL
006087        0242            BVL     EQU     X'0242'
006088                        *       GNB
006089  149E  BD31
006090                        *       DEC
006091  149F  205A
006092                        *       BZF     BVF             MORE TO GO
006093  14A0  5D26
006096                        *       WAIT    DONE
006097  14A1  DFBD
006098  14A2  3100
006099                        *       WAIT
006100  14A3  BD31
006101        14A4            ECP     EQU     $
006102                        *
006103                        *       DEFINED CONSTANTS
006104                        *
006105                        *
006106                        *
006107                        *
006108  14A4  A001            CRC16   DC      Z'A001'
006109  14A5  8408            CCITT   DC      Z'8408'
006110  14A6  0000            LRC8    RESV    1,0
006111  14A7  F010            CRC12   DC      Z'F010'
006112  14A8  FFFF            ALLONE  DC      Z'FFFF'         MINUS ONE OR ALL BITS SET CONSTANT
006113  14A9  0010            C16     DC      X'10'           DECIMAL 16 CONSTANT
006114  14AA  0000            FRST    DC      0               FRST TIME THRU FLAG
006115  14AB  0000            LOOP    DC      0               LOOP COUNTER
006116  14AC  0000            PASS    DC      0               PASS COUNT
006117  14AD  FFFF            ATLT    RESV    16,-1           ACTIVE LINE TABLE
006118  14BD  0000            IMASK   RESV    1,0             MASK OF ACTIVE LINES
006119  14BE  5555            DFLT    DC      =X'5555'        DEFAULT VALUE
006120  14BF  0101            PNOP    DC      =X'0101'        NOP PATTERN
006121  14C0  0000            RNG     DC      0               RANGE FOR NOP BUFFER
006122  14C1  805C            CRLF    DC      Z'805C'
006123  14C2  2020            SPACE   DC      Z'2020'
006124  14C3  0000            MLCADD  DC      0               MLCP ADDRESS
006125  14C4  0000            FW-REV  DC      0               MLCP FIRMWARE REV.
006126                        *
006127                        *
006128                        *
006129                        *               STORAGE CONTANTS
006130                        *
006131                        *
006132                        *
006133                        *
006134  14C5  14C7            ADDS    DC      <DUMMY          ADDSRESS STORAGE
006135  14C6  0400            BUFRNG  DC      X'400'          LENGTH OF SEND, RECEIVE BUFFER
006136  14C7  0000            DUMMY   RESV    1,0
006137  14C8  0000            COUNT   RESV    1,0             COUNTER
006138  14C9  0000            CH-CT   DC      0               CHANNEL COUNT
006139  14CA  266A            ENDSAV  DC      <LST
006140  14CB  0000            TIMEIN  DC      0               INITIAL VALUE FOR PASS TIME
006141  14CC  0000            FLAG    DC      0               FLAG
006142  14CD  0000            TPR     RESV    $AF,0           TEMPORARY REGISTER
006143  14CE  0000            TEMP    RESV    $AF,0
006144  14CF  0000            TEMP1   RESV    $AF,0
006145  14D0  0000            TEMPST  RESV    1,0             STATUS STORED HERE
006146  14D1  0000            SCFLG   RESV    1,0             SCAN FLAG
006147  14D2  0000            CCBFLG  RESV    1,0             SET CCB AFTER RUPT FLAG
006148  14D3  0000            CRCAC   RESV    1,0             CRC ACCUMULATOR
006149  14D4  0000            IRG     RESV    1,0             RANGE INPUT VALUE
006150  14D5  0000            RANGE   RESV    1,0             RANGE VALUE
006151  14D6  0000            RANGE1  RESV    1,0             SECONDARY RANGE
006152  14D7  0000            RANGEW  RESV    1,0             RANGE IN WORDS
006153  14D8  0000            RAMAD   RESV    1,0             RAM ADDRESS IN MLCC
006154  14D9  0000            RAMAD2  RESV    1,0             SECONDARY RAM ADDRESS
006155  14DA  0000            IMGA    RESV    $AF,0           DUMMY CCB ADDRESS
006156  14DB  0000            TSA2    DC      0               0 TO END TRAP SAVE AREA
006157  14DC  1547            SBCK    DC      <RMINST
006158  14DD  0000            SBCCB   DC      0               GETS SBCK WITH BYTES REVERSED
006159  14DE  0000            SBCCP   DC      Z'0000'
006160  14DF  08AA                    DC      Z'08AA'
006161  14E0  5500                    DC      Z'5500'
006162  14E1  0000            MASK    RESV    1,0             MASK WORD
006163  14E2  0000            MASK1   RESV    1,0             SECONDARY MASK WORD
006164  14E3  0000            ERAR    RESV    4,0             ERROR ARRAY
006165  14E7  0000            SAV1    RESV    9+7*$AF,0       SUB-PROGRAM REGS. SAVE AREA
006166  14F7  0000            SAV2    RESV    9+7*$AF,0
006167  1507  0000            SAVN    RESV    9+7*$AF,0
006168  1517  0000            SAV3    RESV    9+7*$AF,0
006169  1527  0000            SAV4    RESV    9+5*$AF,0
006170  1535  0000            SAV6    RESV    2*$AF,0
006171  1537  0000            SAV5    RESV    9+7*$AF,0
006172  1547  0000            RMINST  RESV    1,0             RAM INSTRUCTION TO TEST
006173  1548  0000            RTCHN   DC      0               INTERRUPT RETURN CHANNEL
```

```
006174  1549  0000                    SEC     DC      0                       SECONDS STORED HERE
006175  154A  0000                    TTOT    DC      0                       TICK TOTAL
006176  154B  0000                    INTM    DC      0                       TEMPORARY TIME VALUE
006177  154C  002D                    ERF     DC      45                      ERROR FLAG 0 = SHORT PRINTOUT
006178  154D  0000                    STAT    DC      0                       LAST STATUS INPUT STORED HERE
006179  154E  003C                    HRTZ    DC      =60                     PWR FREQ
006180  154F  2020 2020 454E          DON     TEXT    '     END-OF-PASS       DCMC1$'
        1552  442D 4F46 2D50
              4153 5320 2020
              2020 4443 4D43
              3124
006181                                *
006182                                *       I/O CONTROL WORDS
006183                                *
006184  155C  0009                    CONT1   DC      Z'0009'                 IOLD FUNCTION CODE
006185  155D  000C                    CONT2   DC      Z'000C'                 INPUT RANGE FUNCTION CODE
006186  155E  000F                    CONT3   DC      Z'000F'                 OUTPUT CCB CONTROL FUNCTION CODE
006187  155F  001A                    CONT4   DC      X'1A'                   INPUT NEXT STATUS FUNCTION CODE
006188  1560  000B                    CONT5   DC      Z'000B'                 OUTPUT BYTE INTO LCT FUNCTION CODE
006189  1561  0018                    CONT6   DC      Z'0018'                 INPUT STATUS FUNCTION CODE
006190  1562  0005                    CONT7   DC      X'5'                    OUTPUT CHANNEL CONTROL FUNTION CODE
006191  1563  0001                    CONT9   DC      Z'0001'                 OUTPUT MLCC CONTROL FUNTION CODE
006192  1564  001E                    CONT10  DC      X'1E'                   INPUT LCT STATUS
006193  1565  0003                    CONT11  DC      X'3'                    OUTPUT INTERRUPT CONTROL FUNCTION CODE
006194  1566  001C                    CONT12  DC      X'1C'                   INPUT DATA SET STATUS FC
006195                                *
006196                                *
006197                                *
006198                                *
006199                                *
006200                                *CONSOLE MESSAGES
006201                                *
006202                                *
006203  1567  2020 2044 4C43          MESG1   TEXT    '   DLCP CHANNEL $'
        156A  5020 4348 414E
              4E45 4C20 2400
006204  1570  204C 494E 4520          IDMSG   TEXT    ' LINE $'
        1573  2400
006205  1574  2049 4420 3D20          EQ      TEXT    ' ID =  $'
        1577  2024
006206  1578  2020 2065 6E64          DONE    TEXT    '   END-OF-TEST$'
        157B  2D6F 662D 7465
              7374 2400
006207  1580  4D42 2020 2024          ERMG    TEXT    'MB   $'
006208  1583  444C 4343 204E          ERMG1   TEXT    'DLCC NOT ON THIS CHANNEL$'
        1586  4F54 204F 4E20
              5448 4953 2043
              4841 4E4E 454C
              2400
006209  1590  5241 4E47 4520          TSTA    TEXT    'RANGE TEST  $'
        1593  5445 5354 2020
              2400
006210  1597  4343 4220 424C          TSTB    TEXT    'CCB BLOCK READ TEST $'
        159A  4F43 4B20 5245
              4144 2054 4553
              5420 2400
006211  15A2  4D45 4D4F 5259          TSTC    TEXT    'MEMORY TEST $'
        15A5  2054 4553 5420
              2400
006212  15A9  554E 4156 4149          TSTD    TEXT    'UNAVAILABLE RESOURCE TEST $'
        15AC  4C41 424C 4520
              5245 534F 5552
              4345 2054 4553
              5420 2400
006213  15B7  4C43 5420 4F55          TSTE    TEXT    'LCT OUTPUT/INPUT BYTE TEST $'
        15BA  5450 5554 2F49
              4E50 5554 2042
              5954 4520 5445
              5354 2024
006214  15C5  534F 4654 2049          TSTSI   TEXT    'SOFT INITIALIZE TEST $'
        15C8  4E49 5449 414C
              495A 4520 5445
              5354 2024
006215  15D0  4348 414E 4E45          TSTF    TEXT    'CHANNEL PROGRAM TEST $'
        15D3  4C20 5052 4F47
              5241 4D20 5445
              5354 2024
006216  15DB  494E 5354 5255          TSTG    TEXT    'INSTRUCTION TEST $'
        15DE  4354 494F 4E20
              5445 5354 2024
006217  15E4  4F55 5450 5554          TSTH    TEXT    'OUTPUT INTERRUPT CONTROL TEST $'
        15E7  2049 4E54 4552
              5255 5054 2043
              4F4E 5452 4F4C
              2054 4553 5420
              2400
006218  15F4  494E 4954 4941          TSTJ    TEXT    'INITIALIZE AND CHANNEL RESET TEST $'
        15F7  4C49 5A45 2041
              4E44 2043 4841
              4E4E 454C 2052
              4553 4554 2054
              4553 5420 2400
006219  1606  5354 4F50 2049          TSTK    TEXT    'STOP I/O TEST $'
        1609  2F4F 2054 4553
              5420 2400
006220  160E  5245 4345 4956          TSTL1   TEXT    'RECEIVE TEST $'
        1611  4520 5445 5354
              2024
006221  1615  5452 414E 534D          TSTM1   TEXT    'TRANSMIT TEST $'
        1618  4954 2054 4553
              5420 2400
006222  161D  4441 5441 2049          TSTN    TEXT    'DATA INPUT AND CRC TEST $'
        1620  4E50 5554 2041
              4E44 2043 5243
              2054 4553 5420
              2400
006223  162A  4441 5441 204F          TSTP    TEXT    'DATA OUTPUT TEST $'
        162D  5554 5055 5420
              5445 5354 2024
006224  1633  4649 5253 5420          TSTK    TEXT    'FIRST INTERRUPT TEST $'
        1636  494E 5445 5252
              5550 5420 5445
              5354 2024
```

```
006225  163E  4441 5441 2053      TSTS   TEXT   'DATA SET SCAN TEST $'
        1641  4554 2053 4341
              4E20 5445 5354
              2024
006226  1648  5345 434F 4E44      TSTT   TEXT   'SECOND INTERRUPT TEST $'
        164B  2049 4E54 4552
              5255 5054 2054
              4553 5420 2400
006227                           *
006228                           * DUMPER ROUTINE -USES CHANNEL 2 TO READ
006229                           *
006230  1654  3C02                DMP    LDV    $R3,=2                  USE CHANNEL 2
006231  1655  8752                       CL     =$R2                    RAM ADDRESS
006232  1656  9B80 2BAA                 LAB    $B1,<RTB                CPU ADDRESS
006233  1658  0000                       HLT
006234  1659  AF00 1661                  STR    $R2,<DMP2
006235  165B  9F80 165F                  STB    $B1,<DMP1
006236  165D  9380 1098                  LNJ    $B1,<RDATA              READ
006237  165F  2BAA                DMP1   DC     <RTB                    TO HERE
006238  1660  0200                       DC     X'200'                  RANGE
006239  1661  0000                DMP2   DC     X'0'                    RAM ADDRESS
006240  1662  8000                       DC     Z'8000'
006241                           *
006242  1663  0FF1                       B      >DMP                    DO NEXT
006243        1664                V$LB   EQU    $
006244                                     IFZ    ($AF-2),LAF99
006245  1EBB                             ORG    ZERO+X'1EBB'
006248        1EBB                SDB    EQU    $                       SEND BLOCK AREA
006249                                     IFZ    ($AF-2),LAF98
006250  2BAA                             ORG    ZERO+X'2BAA'
006253        2BAA                RTB    EQU    $                       RETURN BLOCK AREA
006257  25AA                SAF9   ORG    ZERO+X'25AA'
006258        25AA                LAF2   EQU    $
006259                           * INTERRUPT SAVE AREA FOLLOWS. THIS AREA WILL BE OVER WRITTEN BY
006260                           * DATA DURING DATA TRANSFERS BUT THE FOLLOWING SAVE AREAS ARE
006261                           * RE-CREATED BEFORE TESTING INTERRUPTS
006262                           *
006263                           *
006264                           * INTERRUPT SAVE AREAS - 16 FOR MLCC
006265                           *
006266                           * CHANNEL 1
006267  25AA  0000               ISI    RESV   $AF,0                    TRAP SAVE POINTER
006268  25AB  0000                      RESV   1,0                      CHAN,LEVEL
006269  25AC  7884                      RESV   1,X'7884'                SAVE R1,R2,R3,R4,I,B5
006270  25AD  0000                      RESV   1,0                      RFU
006271  25AE  1020                      DC     <IHD1
006272  25AF  FFFF                      DC     -1                       MAKE SURE PRIVLEGE BIT IS SI
006273  25B0  0000                      RESV   5+$AF,0
006274                           * CHANNEL 2
006275  25B6  0000                      RESV   $AF,0                    TRAP SAVE POINTER
006276  25B7  0000                      RESV   1,0                      CHAN,LEVEL
006277  25B8  7884                      RESV   1,X'7884'                SAVE R1,R2,R3,R4,I,B5
006278  25B9  0000                      RESV   1,0                      RFU
006279  25BA  1020                      DC     <IHD1
006280  25BB  FFFF                      DC     -1                       SET PRIVILEGE BIT
006281  25BC  0000                      RESV   5+$AF,0
006282                           * CHANNEL 3
006283  25C2  0000                      RESV   $AF,0                    TRAP SAVE POINTER
006284  25C3  0000                      RESV   1,0                      CHAN,LEVEL
006285  25C4  7884                      RESV   1,X'7884'                SAVE R1,R2,R3,R4,I,B5
006286  25C5  0000                      RESV   1,0                      RFU
006287  25C6  1020                      DC     <IHD1
006288  25C7  FFFF                      DC     -1                       SET PRIVILEGE
006289  25C8  0000                      RESV   5+$AF,0
006290                           * CHANNEL 4
006291  25CE  0000                      RESV   $AF,0                    TRAP SAVE POINTER
006292  25CF  0000                      RESV   1,0                      CHAN,LEVEL
006293  25D0  7884                      RESV   1,X'7884'                SAVE R1,R2,R3,R4,I,B5
006294  25D1  0000                      RESV   1,0                      RFU
006295  25D2  1020                      DC     <IHD1
006296  25D3  FFFF                      DC     -1                       SET PRIVILEGE BIT
006297  25D4  0000                      RESV   5+$AF,0
006298                           * CHANNEL 5
006299  25DA  0000                      RESV   $AF,0                    TRAP SAVE POINTER
006300  25DB  0000                      RESV   1,0                      CHAN,LEVEL
006301  25DC  7884                      RESV   1,X'7884'                SAVE R1,R2,R3,R4,I,B5
006302  25DD  0000                      RESV   1,0                      RFU
006303  25DE  1020                      DC     <IHD1
006304  25DF  FFFF                      DC     -1                       SET PRIVILEGE BIT
006305  25E0  0000                      RESV   5+$AF,0
006306                           * CHANNEL 6
006307  25E6  0000                      RESV   $AF,0                    TRAP SAVE POINTER
006308  25E7  0000                      RESV   1,0                      CHAN,LEVEL
006309  25E8  7884                      RESV   1,X'7884'                SAVE R1,R2,R3,R4,I,B5
006310  25E9  0000                      RESV   1,0                      RFU
006311  25EA  1020                      DC     <IHD1
006312  25EB  FFFF                      DC     -1                       SET PRIVILEGE BIT
006313  25EC  0000                      RESV   5+$AF,0
006314                           * CHANNEL 7
006315  25F2  0000                      RESV   $AF,0                    TRAP SAVE POINTER
006316  25F3  0000                      RESV   1,0                      CHAN,LEVEL
006317  25F4  7884                      RESV   1,X'7884'                SAVE R1,R2,R3,R4,I,B5
006318  25F5  0000                      RESV   1,0                      RFU
006319  25F6  1020                      DC     <IHD1
006320  25F7  FFFF                      DC     -1                       SET PRIVILEGE BIT
006321  25F8  0000                      RESV   5+$AF,0
006322                           * CHANNEL 8
006323  25FE  0000                      RESV   $AF,0                    TRAP SAVE POINTER
006324  25FF  0000                      RESV   1,0                      CHAN,LEVEL
006325  2600  7884                      RESV   1,X'7884'                SAVE R1,R2,R3,R4,I,B5
006326  2601  0000                      RESV   1,0                      RFU
006327  2602  1020                      DC     <IHD1
006328  2603  FFFF                      DC     -1                       SET PRIVILEGE BIT
006329  2604  0000                      RESV   5+$AF,0
006330                           * CHANNEL 9
006331  260A  0000                      RESV   $AF,0                    TRAP SAVE POINTER
006332  260B  0000                      RESV   1,0                      CHAN,LEVEL
006333  260C  7884                      RESV   1,X'7884'                SAVE R1,R2,R3,R4,I,B5
006334  260D  0000                      RESV   1,0                      RFU
006335  260E  1020                      DC     <IHD1
006336  260F  FFFF                      DC     -1                       SET PRIVILEGE BIT
006337  2610  0000                      RESV   5+$AF,0
006338                           * CHANNEL 10
```

```
006339  2616  0000                      RESV  $AF,0          TRAP SAVE POINTER
006340  2617  0000                      RESV  1,0           CHAN,LEVEL
006341  2618  7884                      RESV  1,X'7884'     SAVE R1,R2,R3,R4,I,B5
006342  2619  0000                      RESV  1,0           RFU
006343  261A  1020                      DC    <IHD1
006344  261B  FFFF                      DC    -1            SET PRIVILEGE BIT
006345  261C  0000                      RESV  5+$AF,0
006346                            * CHANNEL 11
006347  2622  0000                      RESV  $AF,0         TRAP SAVE POINTER
006348  2623  0000                      RESV  1,0           CHAN,LEVEL
006349  2624  7884                      RESV  1,X'7884'     SAVE R1,R2,R3,R4,I,B5
006350  2625  0000                      RESV  1,0           RFU
006351  2626  1020                      DC    <IHD1
006352  2627  FFFF                      DC    -1            SRT PRIVILEGE BIT
006353  2628  0000                      RESV  5+$AF,0
006354                            * CHANNEL 12
006355  262E  0000                      RESV  $AF,0         TRAP SAVE POINTER
006356  262F  0000                      RESV  1,0           CHAN,LEVEL
006357  2630  7884                      RESV  1,X'7884'     SAVE R1,R2,R3,R4,I,B5
006358  2631  0000                      RESV  1,0           RFU
006359  2632  1020                      DC    <IHD1
006360  2633  FFFF                      DC    -1            SET PRIVILEGE BIT
006361  2634  0000                      RESV  5+$AF,0
006362                            * CHANNEL 13
006363  263A  0000                      RESV  $AF,0         TRAP SAVE POINTER
006364  263B  0000                      RESV  1,0           CHAN,LEVEL
006365  263C  7884                      RESV  1,X'7884'     SAVE R1,R2,R3,R4,I,B5
006366  263D  0000                      RESV  1,0           RFU
006367  263E  1020                      DC    <IHD1
006368  263F  FFFF                      DC    -1            SET PRIVILEGE BIT
006369  2640  0000                      RESV  5+$AF,0
006370                            * CHANNEL 14
006371  2646  0000                      RESV  $AF,0         TRAP SAVE POINTER
006372  2647  0000                      RESV  1,0           CHAN,LEVEL
006373  2648  7884                      RESV  1,X'7884'     SAVE R1,R2,R3,R4,I,B5
006374  2649  0000                      RESV  1,0           RFU
006375  264A  1020                      DC    <IHD1
006376  264B  FFFF                      DC    -1            SET PRIVILEGE BIT
006377  264C  0000                      RESV  5+$AF,0
006378                            * CHANNEL 15
006379  2652  0000                      RESV  $AF,0         TRAP SAVE POINTER
006380  2653  0000                      RESV  1,0           CHAN,LEVEL
006381  2654  7884                      RESV  1,X'7884'     SAVE R1,R2,R3,R4,I,B5
006382  2655  0000                      RESV  1,0           RFU
006383  2656  1020                      DC    <IHD1
006384  2657  FFFF                      DC    -1            SET PRIVILEGE BIT
006385  2658  0000                      RESV  5+$AF,0
006386                            * CHANNEL 16
006387  265E  0000                      RESV  $AF,0         TRAP SAVE POINTER
006388  265F  0000                      RESV  1,0           CHAN,<LEVEL
006389  2660  7884                      RESV  1,X'7884'     SAVE R1,<R2,<R3,<R4,<I,B5
006390  2661  0000                      RESV  1,0           RFU
006391  2662  1020                      DC    <IHD1
006392  2663  FFFF                      DC    -1            SET PRIVILEGE BIT
006393  2664  0000                      RESV  5+$AF,0
006394        266A                LS1   EQU   $
006395                            *
006396                            * INTERRUPT SAVE AREA - LEV 0
006397                            *
006398  266A  0000                      RESV  $AF,0         TRAP SAVE POINTER
006399  266B  0000                TOPSAV RESV 1,0           CHAN, LEVEL
006400  266C  FFFF                      DC    Z'FFFF'       SAVE ALL REG
006401  266D  0000                      RESV  1,0           RFU
006402  266E  14C7                      DC    <DUMMY        SUPPLIED AT RUPT TIME
006403  266F  FFFF                      DC    -1            SET PRIVILEGE BIT
006404  2670  0000                      DC    0             FILLED AT SAVE TIME
006405  2671  0000                      RESV  8+7*$AF,0     ROOM FOR SAVE
006406                            *
006407                            * COUNT TABLE FOR INTERRUPTS
006408                            *
006409  2680  0000                INMB  RESV  16,0
006410                            *
006411                            * STATUS TABLE FOR INTERRUPTS
006412                            *
006413  2690  0000                IST   RESV  16,0
006414                            *
006415                            * LEVEL TABLE FOR INTERRUPTS
006416                            *
006417  26A0  0000                ILV   RESV  16,0
006418                            *
006419                            * PRIORITY TABLE FOR INTERRUPTS
006420  26B0  0000                PRIST RESV  16,0
006421                            *
006422                            * READ FIRMWARE REV NUMBER
006423                            *
006424  26C0  8F00 1EBB           PREV  SAVE  <SDB,=Z'0008'   SAVE B4
      26C2  0008
006425  26C3  C380 10FC                 LNJ   $B4,<GENITZ     GENERAL INITAILIZE
006426  26C5  9380 1098                 LNJ   $B1,<RDATA      BLOCK READ
006427  26C7  2BAA                      DC    <RTB            TO HERE
006428  26C8  0002                      DC    2               RANGE
006429  26C9  0000                      DC    0
006430  26CA  0000                      DC    0               EVEN BYTE START
006431                            *
006432                            *
006433                                  CALL  ZV$TC,FREV      PRINT FIRMWARE REV
      26CB  FBC0 0003
      26CD  D380 0000           X
      26CF  0F80
      26D0  26F8
006434  26D1  9800 2BAA                 LDR   $R1,<RTB        GET REV NUMBER
006435  26D3  9570 00E0                 AND   $R1,=Z'00E0'
006436  26D5  1045                      SOR   $R1,5           ALIGN INTERIM REV NUMBER
006437  26D6  9F00 14CE                 STR   $R1,<TEMP
006438  26D8  1900 26E0                 BEZ   $R1,<SIT12      BYPASS INTERIM REV IF 0
006439                                  CALL  ZV$TD,TEMP      TYPE INTERIM
      26DA  FBC0 0003
      26DC  D380 0000           X
      26DE  0F80
      26DF  14CE
006440                            *
006441  26E0  9800 2BAA           SIT12 LDR   $R1,<RTB        GET REV NUMB
006442  26E2  9570 001F                 AND   $R1,=Z'001F'    STRIP TO REV
```

```
006443  26E4  9F00 14C4                    STR    $R1,<FW-REV          PUT BACK
006444                                     CALL   ZV$TD,FW-REV         PRINT NUMBER
        26E6  FBC0 0003
        26E8  D380 0000        X
        26EA  0F80
        26EB  14C4
006445  26EC  6C01                         LDV    $R6,=1
006446  26ED  D800 14C4                    LDR    $R5,<FW-REV
006447  26EF  D956                         CMR    $R5,=$R6
006448  26F0  0284                         BGE    >BLAP
006449  26F1  F380 125C              WFW    LNJ    $B7,<ERROR           WRONG FIRMWARE REV,
006450                               *                                  REV C OF PROGRAM DOESN'T SUPPORT
006451  26F3  0FFE                         B      >WFW
006452  26F4  8F80 1EBB              BLAP   RSTR   <SDB,=Z'0008'        RESTORE B4
        26F6  0008
006453  26F7  8384                         JMP    $B4
006454  26F8  2020 2046 4952        FREV   TEXT   '   FIRMWARE REV.$'
        26FB  4D57 4152 4520
              5245 562E 2400
006455  2701  2020 2020 2044        MESG   TEXT   '     DLCP  TEST '
        2704  4C43 5020 2054
              4553 5420
006456                                     IFZ    ($AF-2),LAF8
006457  2709  4443 4D43 312C              TEXT   'DCMC1, SAF-E  '
        270C  2053 4146 2045
              2020
006460  2710  204D 4159 2020        DATE   TEXT   ' MAY  11,  1978$'
        2713  3131 2C20 2031
              3937 3824
006461  2718  2020 2050 4F57        MESG5  TEXT   '   POWER FREQ (HZ) $'
        271B  4552 2046 5245
              5120 2848 5A29
              2024
006462  2722  434F 5059 5249              TEXT   'COPYRIGHT 1976 BY HONEYWELL INFORMATION SYSTEMS INC.'
        2725  4748 5420 3139
              3736 2042 5920
              484F 4E45 5957
              454C 4C20 494E
              464F 524D 4154
              494F 4E20 5359
              5354 454D 5320
              494E 432E
006463                               *
006464                               *
006465                               *
006466                               *
006467  1664                               ORG    V$LB
006468  1664  0100                         END    DCMC1,STRT
0000  ERR COUNT
   TITLE DCMC1  'REV. A'
        $AF      406B   507C   524B   564B   568B,  575B   591B   596B   600B   606B
                 610B   622B   641    682B   692B   925B   930B  1095B  1110B  1295
                1320   1349B  1358B  1517B  1537B  1562B  1566B  1570B  1607B  1639B
                1655B  1711B  1747B  1777B  1782B  1786B  1838B  1846B  2114B  2166B
                2177B  2251B  2284B  2297B  2432B  2441B  2457B  2608C  2609C  2679
                2974B  3117B  3143B  3203B  3214B  3282B  3286B  3301B  3332B  3347B
                3357B  3426C  3529B  3540   3549   3556   3567   3573   3617   3637
                3657   3658C  3659   3660C  3661   3662C  3722C  3726   3916   3932B
                3935B  4005B  4016   4018   4062B  4122C  4145   4189B  4252B  4257B
                4269B  4369C  4428   4429C  4433   4434   4438   4444   4445   4449
                4451   4456C  4459   4465   4473   4474   6142   6143   6144   6155
                6165   6166   6167   6168   6169   6170   6171   6244   6246   6249
                6251   6254   6256   6267   6273   6275   6281   6283   6289   6291
                6297   6299   6305   6307   6313   6315   6321   6323   6329   6331
                6337   6339   6345   6347   6353   6355   6361   6363   6369   6371
                6377   6379   6385   6387   6393   6398   6405   6456   6458
        $B1      502B   755B   769B   777B   784B   789B   796B   799B   806B   832B
                 840B   843B   850B   863B   870B   873B   880B   887B   901B  1032
                1036C  1041C  1065B  1091   1093   1145B  1153B  1169   1171   1192B
                1201B  1271B  1278B  1396B  1403B  1410B  1416B  1422B  1428B  1437B
                1462B  1528B  1600B  1602B  1618B  1644B  1645B  1646B  1659B  1660B
                1673B  1680B  1692B  1733B  1770B  1789B  1800B  2090   2091C  2141B
                2233   2234C  2308B  2372B  2488   2495C  2498C  2500   2502C  2543B
                2558B  2568B  2577B  2692B  2717B  2792B  2906B  2922B  2951   2952C
                2982B  3007   3008C  3020B  3077B  3175B  3181   3182C  3239B  3255
                3256C  3308B  3373B  3410B  3439B  3469B  3474B  3515B  3702   3707
                3708C  3712   3724B  3745   3746   3748   3755B  3770B  3778   3781
                3797   3823   3907   3908   3910   3912   3938B  3953   3955   3956
                3958   4290   4296C  4444   4449   4450C  4478C  6232   6235C  6236B
                6426B
        $B2      457B   458B   459B   460B   461B   462B   463B   464B   465B   466B
                 467B   468B   469B   470B   471B   472B   473B   474B   586B   650B
                 732B   917B   944B   961B  1118B  1134B  1215B  1263B  1351B  1381B
                1509B  1553B  1814B  1964B  2055B  2352   2355C  2359C  2363C  2411
                2536B  2709B  2809B  2905B  3071B  3521B  3630   3634C  3638   3640C
                3648   3649C  3651   3652C  3653   3655C  3658C  3660C  3662C  3748
                3749   4043   4046   4245   4251   4433   4437C  4438   4440C  4464C
                4468C  4474   4477C
        $B3      757    760C  1042B  1046   1047   1075   1076   1078C  1090   1092
                1161B  1167B  1285B  1289B  1316B  1318B  1334B  1360B  1445B  1451B
                1595B  1688B  1694B  1702B  1817B  2361   2362C  2366   2367C  2379B
                2391B  2426   2430   2552   2600   2602   2604   2605C  2613B  2643
                2644C  2650B  2691   2727   2734B  2758B  2785   2910B  2960B  3029B
                3090B  3154B  3193B  3224B  3231B  3399   3400C  3485B  3493B  3497B
                3633   3634C  3699   3702   3703   3705C  3706   3708C  3710C  3712
                3713   3713   3714   3749   3751C  3778   3785B  3797   3798   3826B
                3977   3978   3982B  4011B  4018   4024C  4124   4144C  4151   4168C
                4289   4291   4293   4443   4450C  4455   4456C  4478C
        $B4      361    362C   363    364C   404B   425B   504B   510B   517B   528B
                 536B   554B   558B   562B   585B   589B   594B   604B   620B   639B
                 649B   667B   674B   680B   687B   689B   707B   726B   731B   914B
                 916B   923B   928B   933B   937B   939B   957B   960B   968B   998B
                1051B  1052B  1081B  1105B  1130B  1133B  1168B  1172B  1173B  1182B
                1202B  1207B  1214B  1260B  1262B  1288B  1297B  1301B  1302B  1326B
                1340B  1341B  1343B  1356B  1380B  1384B  1448B  1454B  1508B  1513B
                1515B  1552B  1555B  1557B  1560B  1573B  1579B  1587B  1605B  1609B
                1637B  1653B  1664B  1674B  1678B  1695B  1696B  1703B  1705B  1708B
                1762B  1768B  1769B  1775B  1794B  1811B  1813B  1816B  1820B  1826B
                1830B  1834B  1843B  1851B  1854B  1872B  1873B  1876B  1877B  1881B
                1895B  1908B  1922B  1935B  1950B  1963B  1985B  1999B  2013B  2027B
                2041B  2054B  2077   2078   2084   2087   2090   2093   2112B  2128B
                2130B  2151B  2162B  2174B  2189B  2190B  2201B  2216   2217   2222
                2226   2233   2236   2249B  2257B  2259B  2264B  2276B  2294B  2325B
```

```
                2326B   2336B   2352    2353    2361    2364    2366    2368C   2385B   2393B
                2395    2397B   2411    2412    2446B   2491    2492    2494    2505B   2516B
                2526B   2594B   2637B   2653B   2656B   2671B   2686B   2706B   2708B   2732B
                2738B   2743B   2748B   2753B   2762B   2771B   2775B   2789B   2808B   2903B
                2904B   2907B   2908B   2933B   2939B   2943B   2946B   2953B   2972B   2997B
                3004B   3009B   3025B   3070B   3072B   3073B   3113B   3114B   3115B   3126B
                3136B   3137B   3141B   3163B   3168B   3172B   3174B   3183B   3201B   3209B
                3211B   3221B   3222B   3238B   3242B   3251B   3257B   3269B   3279B   3280B
                3294B   3298B   3324C   3330B   3334C   3335B   3343C   3345B   3349C   3350B
                3353C   3355B   3359C   3360B   3383B   3409B   3414B   3420B   3481B   3487B
                3501B   3506B   3527B   3532B   3788B   3829B   3915B   3918B   3925B   3929B
                3930B   3961B   3964B   3992B   4003B   4027B   4049    4051B   4057B   4060B
                4064B   4066B   4149B   4187B   4193B   4210B   4219    4220    4222B   4228B
                4245    4246    4247C   4250B   4255B   4259C   4261B   4265C   4267B   4272C
                4273B   4289    4290    4299B   4313    4314    4318B   4325B   4333B   4336B
                4346B   4371B   4377B   4403B   4410    4413C   4414B   4415    4419B   4432
                4439    4464C   4476    4477C   6425B   6453B
$B5              432     436     438C   2182B   2318B   2368C   2382    2383    2395    2423
                2424    2459B   2520    2521C   2597    2598C   2602    2607C   2608C   2609C
                2611    2612C   2679    3324C   3334C   3343C   3349C   3353C   3359C   3629
                3630    3631    3633    3637    3640C   3646    3649C   3654    3655C   3657
                3658C   3659    3660C   3661    3662C   3673    3674C   3881    3882    3885C
                3886C   3907    3909C   3953    3954C   4016    4018    4022C   4024C   4247C
                4259C   4265C   4272C   4308    4368    4369C   4436    4437C
$B6              739B    741B    743B    745B    883B   2427    2429    2438    2551    2562
$B7             2573    2728    2786    4115    4428    4429C   4431    4440C   4467    4468C
                 407B    520B    525B    531B    539B    549B    565B    569B    576B    592B
                 597B    601B    607B    611B    624B    683B    693B    703B    899B    924
                 926B    931B    943B   1056B   1096B   1112B   1175B   1186B   1306B   1311B
                1350B   1359B   1474B   1518B   1538B   1563B   1567B   1571B   1608B   1629B
                1640B   1656B   1712B   1748B   1778B   1783B   1787B   1839B   1847B   2167B
                2178B   2195B   2285B   2298B   2331B   2433B   2442B   2458B   2666B   2683B
                2768B   2805B   2975B   3118B   3144B   3204B   3215B   3283B   3287B   3302B
                3333B   3348B   3358B   3530B   3589B   3596B   3766B   3848B   3856B   3860B
                3864B   3873B   3933B   3936B   4006B   4063B   4190B   4225B   4253B   4258B
                4270B   4328B   4428    4472B   6449B
$IV             3843    3849
$R1              349     350C    353     354C    359     360     365     368     369     391
                 394     402     424     434     435C    437     438C    446     447C    482
                 483     489     490     499     500C    506     507C    508C    560     572B
                 587     615B    635     636C    641     642C    643C    722     723C    737C
                 829     830B    918C    974C    977     978C    979C    984C    985     986C
                 993     994C    995     996C   1030    1036C   1038    1041C   1047    1048C
                1049    1050    1089C   1092    1093    1097C   1098    1126    1127C   1171
                1199    1200C   1266C   1294    1296B   1319    1321B   1387C   1575    1576C
                1625    1631B   1871    1875    1892    1893C   1906    1907C   1920    1921C
                1933    1934C   1948    1949C   1961    1962C   1982    1983C   1997    1998C
                2011    2012C   2025    2026C   2039    2040C   2052    2053C   2075    2076C
                2077    2078    2079C   2080C   2081C   2082C   2084    2085C   2087    2088C
                2093    2094C   2095C   2096C   2098    2107    2108    2109C   2188    2216
                2217    2218C   2219C   2220C   2221C   2222    2223C   2226    2227    2230C
                2236    2237C   2238    2240    2241C   2242C   2243    2244    2245    2246C
                2324    2350    2351C   2364    2365C   2382    2413    2414C   2423    2429
                2430    2438    2494    2495C   2498C   2499    2503B   2531    2532C   2533C
                2534    2538C   2599    2600    2636C   2642    2643    2670C   2705C   2914
                2915C   2917    2918C   2931C   2983    2987    2989    2990C   2992    2993C
                2995    2996C   3003C   3021    3022B   3110    3112    3122    3124    3125C
                3149    3150C   3156    3157    3159C   3160    3162    3169    3171    3176
                3177C   3208    3240    3241C   3243    3244C   3245    3246C   3250    3266
                3268    3291    3293    3325    3326    3327C   3328    3331    3346    3364
                3368    3371    3372C   3417C   3434    3435    3436B   3438C   3452    3455
                3456C   3460B   3492    3502B   3639    3641B   3645    3650B   3663C   3664C
                3665C   3666C   3667C   3668C   3669C   3670    3671C   3700C   3703    3720
                3721C   3722C   3747    3751C   3753B   3779C   3781    3782C   3798    3800C
                3801    3803    3805    3807    3809    3811    3813    3815    3817    3819
                3821    3823    3841    3842    3861    3912    3913C   3914    3956    3957C
                3958    3959C   3960    4000    4001B   4004    4019    4020    4022C   4116C
                4132C   4134    4136B   4155C   4157    4160    4251    4293    4294    4295
                4309C   4310C   4311    4312C   4313    4314    4315C   4316C   4331    4332
                4339    4340C   4341    4342    4353    4354    4361    4362    4363    4366
                4367    4370C   4372C   4373C   4374    4375C   4379    4381B   4382    4392C
                4410    4411C   4412C   4416    4417C   4426    4469B   6434    6435    6436
                6437C   6438B   6441    6442    6443C
$R2              431     436     438C    439B    555     563     570C    574     582     598
                 613C    750     975     976     980     985     987B    988C   1033C   1034
                1035    1036C   1040C   1041C   1045    1047    1057C   1058    1074    1076
                1078C   1079B   1170C   1171    1176    1558C   1565    1626    1627    1630
                1744    1745    1749C   1750C   1752    1754    1756    1758    1760    1867
                1869    1871    2225C   2229    2231C   2356C   2357C   2358    2359C   2383
                2384C   2389    2424    2425C   2434B   2491    2495C   2496C   2501    2502C
                2522    2523C   2596    2597    2610    2611    3404    3429C   3430B   3433
                3434    3447    3448B   3453C   3457    3459    3462C   3463C   3464C   3745
                3751C   3761    3768B   3781    3783B   3789    3823    3824B   3830    3868
                3870    3910    3911C   4015C   4019    4023C   4025    4042    4046    4049
                4120C   4134    4135    4137    4141    4144C   4147C   4157    4158    4159
                4162    4166    4168C   4169    4172    4178    4246    4256    6231C   6234C
$R3              391     392     393     395C    396C    408C    409B    411     412C    414
                 420     421C    422     509C    546C    557C    577C    584C    626C    638C
                 651C    652C    653C    654C    656     659C    660B    661C    663C    672
                 678     694     709C    730C    734B    735C    746C    764C    765C    766B
                 768C    776C    788C    798C    814     828C    842C    862C    872C    882
                 910     911C    913C    954     955C    959C    962C    963B    964C   1104B
                1114    1115C   1116    1132C   1141C   1142B   1143C   1166    1188   1189B
                1190C   1206C   1212    1213C   1256    1257C   1259C   1264B   1265C   1267C
                1325C   1333C   1375    1376C   1378C   1382C   1383C   1385B   1386C   1456
                1460C   1475    1476C   1504    1505C   1507C   1520    1524C   1525B   1526C
                1534    1539C   1548    1549C   1551C   1580    1611C   1612    1613    1614C
                1615    1616    1632    1713C   1726    1729C   1730B   1731C   1739    1751B
                1808    1809C   1812C   1856C   1868B   2111C   2116B   2117C   2119    2123C
                2139C   2147C   2196C   2248C   2253B   2254C   2256C   2302    2307C   2314C
                2332C   2369C   2370B   2371C   2378    2448    2449B   2492    2493C   2497B
                2513    2514C   2524C   2528B   2529C   2539C   2550C   2579    2701    2702C
                2704C   2710B   2711C   2777    2791C   2807    2810B   2811C   2898    2899C
                2901C   2915C   2918C   2920C   2921    2927    2990C   2993C   2996C   3036C
                3066    3067C   3069C   3075C   3076    3083    3111B   3123B   3161B   3177C
                3241C   3244C   3267B   3292B   3312C   3365C   3366B   3367C   3369    3382
                3392    3393C   3403C   3425C   3455    3458B   3486C   3746    3750B   3752
                3760C   3762    3764    3767C   3843    3844    3845    3849    3850   3851C
                3852    3853    3861    3870    3885C   3888C   3979    3980B   3985    3989
                3999C   4000    4008C   4009    4044    4046    4048C   4056C   4119C   4124
                4151    4164    4165    4167    4170    4171    4173    4207    4208    4209
                4427C   4443    4446    4454C   4455    4463C   4465    6230
$R4              370     390     402     403     405     424     426     433     436     437
```

		513	514	515C	516	518	522	523	527	529	535
		537	561	563	566	567	573	574	588	590	593
		595	603	605	619	621	645	646C	665	673	679
		681	699	728	729C	738C	740	742	744	760C	761
		785	786C	814	815	816C	817C	818	819C	820C	821
		822C	823C	824C	825	826C	858	859	860C	895	922
		924	927	929	976	977	978C	983	988C	1031	1033C
		1035	1037C	1060	1061B	1062	1063	1064C	1076	1077	1078C
		1293C	1347C	1355	1357	1388	1389C	1447	1456	1457	1458
		1459C	1514	1516	1520	1521	1522	1523C	1556	1558C	1559
		1561	1569	1588	1589C	1590C	1591C	1592C	1593C	1594C	1604
		1606	1636	1638	1652	1654	1671	1672C	1726	1727	1728C
		1774	1776	1780	1781	1793	1797C	1798	1819	2105	2106C
		2119	2120	2121	2122C	2302	2303C	2304	2305	2306C	2353
		2356C	2412	2489	2490C	2498C	2635	2641	2642	2655	2658
		2669	2737	2742	2747	2752	2770	2930	2945	2947C	3002
		3024	3329	3331	3344	3346	3354	3356	3369	3370	3371
		3401	3402C	3405	3526	3528	3704	3705C	3706	3709	3710C
		3787	3789	3828	3830	3851C	3867	3868	3908	3955	3984
		3993	4002	4004	4059	4061	4115	4128	4130	4137	4162
		4186	4188	4208	4221	4223	4248	4249	4251	4254	4256
		4266	4268	4291	4292	4295	4296C	4324	4326	4376	4383
		4386	4387	4397							
	$R5	369	370	371	372	378C	390	395C	397	405	408C
		416	417	418C	426	537	540	595	599	605	609
		621	647	648C	656	657	658	662	664C	691	700
		852	853	896	919	920C	924	941	973	981B	1054
		1080	1092	1094	1109	1174B	1184	1304	1309	1348	1469
		1470	1472	1535	1536	1627	1628B	1638	1710	1745	1746
		1781	1785	1837	1845	1874C	1878	1879	2101	2102C	2103C
		2165	2175	2176	2191	2193	2281	2283	2295	2296	2327
		2329	2354	2355C	2359C	2429	2431	2438	2440	2456	2553
		2561	2571	2657C	2658	2661	2680	2729	2766	2784	2799
		2600	2801	2964	2973	3116	3127	3128	3129C	3130C	3131
		3132	3135C	3142	3202	3213	3281	3285	3300	3379	3380
		3381C	3396	3397C	3418	3424B	3432	3443C	3444	3446C	3519
		3528	3931	3934	3977	3993	4123B	4124	4132C	4133	4141
		4142	4143C	4151	4154	4155C	4156	4169	4174	4175C	4178
		4179C	4188	4191C	4219	4220	4223	4248	4251	4268	4271
		4288C	4297B	4319	4326	4386	4397	4399C	6446	6447	
	$R6	503	518	540	544	598	599	608	609	690	691
		701	751	752C	758	759C	762B	897	1048C	1053	1054
		1093	1094	1108C	1109	1183	1184	1268	1269C	1292	1304
		1308	1309	1346	1348	1471	1472	1510	1511	1512	1516
		1536	1709	1710	1743C	1746	1784	1785	1836	1844C	1845
		1866	1870	1875	2163	2164	2165	2168C	2170	2176	2192C
		2193	2275C	2278	2282	2283	2287C	2288	2296	2328C	2329
		2430	2431	2439	2440	2450	2453	2455	2456	2518	2519C
		2554	2555C	2556	2563	2564C	2565	2566C	2567	2574	2575C
		2576	2648C	2064B	2681	2730	2777	2778	2779	2780C	2787
		2803	3212	3213	3299	3300	3978	3987B	3988	4121C	4122C
		4138B	4145	4163B	4388	4389	6445	6447			
	$R7	556	567	571	583	608	614	702	898	1750C	2557
		2634	2645B	2647	2667	2674B	2682	2690	2731	2788	3125C
		3128	3132	3913C	3924B	3959C	3988	3989	3990B	4126	4127C
		4146B									
3421	ADD	3400C	3426C	3429C							
6134	ADDS	2521C	2597	2598C	2605C	2611					
1768	ADV2	1602B	1646B								
433	ALL	439B									
6112	ALLONE	386	3692								
6117	ATLT	386	395C	408C	4043						
2968	BKHR	2957B									
457	BKK1	440B	451B								
467	BKK10										
456	BKK3	456	485B								
459	BKK4										
470	BKK4A										
460	BKK4B										
471	BKK5										
463	BKK7										
474	BKK7A										
464	BKK7B										
468	BKK7C										
469	BKK7D										
473	BKK9										
466	BKK9B										
6452	BLAP	6448B									
708	BLUG	706B									
1726	BRLCLR	1579B	1674B								
4508	BT1	4502	4503	4504							
4566	BT11										
4571	BT12	4563	4564	4568							
4583	BT17	4574	4575	4577							
4588	BT18	4580	4581	4585							
4614	BT1D										
4815	BT4B	4783	4784	4786	4797	4798	4799	4833	4834	4838	
4751	BT4C	4718	4719	4720	4733	4734	4736	4768	4769	4770	
4674	BT4D	4642	4643	4645	4656	4657	4658	4704	4705	4707	
4629	BT4F	4623	4624	4626							
5117	BT4G	5109	5110	5114							
5112	BT4H	5121	5122	5123							
4836	BT4J										
4525	BT5	4557	4558	4560							
4529	BT6	4521	4522	4523							
5012	BTA6	4887	4888	4889							
4883	BTA7	4875	4876	4877							
5043	BTA9	5025	5026	5028	5060	5061	5062	5075	5076	5078	
4679	BTAC	4671	4672	4676							
4549	BTB	4533	4534	4538							
5047	BTB5	5039	5040	5041							
4879	BTB7	4898	4899	4904							
4553	BTC	4545	4546	4547							
4755	BTRC	4747	4748	4749							
4820	BTXC	4812	4813	4817							
4859	BTYD	5166	5167	5168							
6135	BUFRNG	1138	2127								
6055	BVF	6094	6095	6097							
6018	BVFF	6009	6010	6011							
6080	BVK	6072	6073	6074							
6087	BVL	6076	6077	6078							
6068	BVR	6059	6060	6061							
6005	BVT	6044	6045	6047							

```
6030  BVTT      6022    6023    6024
6037  BWAIT     6026    6027    6028
2335  BWEND     2252B
2269  BWLCT     2234C
6113  C16        386    3679    3683    3688    3692
1621  CADD      1614C
 669  CBRT1      664C    699
 663  CBRT3      660B
6147  CCBFLG    3665C
3414  CCBLP     3430B
 635  CCBRT      458B
2271  CCBSAV    2230C   2277    2279    2286    2451
6109  CCITT     1138
4207  CGSCH      404B    425B    517B    528B    536B    562B    589B    594B    604B    620B
                 680B    923B    928B   1356B   1515B   1557B   1560B   1605B   1637B   1653B
                1775B   2656B   2946B   3330B   3345B   3355B   3527B   3788B   3829B   3992B
                4003B   4060B   4187B   4222B   4250B   4255B   4267B   4325B   4377B
      CH        4854    4855    5150    5151    5166    5167    5181    5182    5437    5438
                5450    5451    5705    5706    5716    5717
6138  CH=CT      387C
3441  CHAN      3438C
4218  CHCT       933B   1202B   1207B   1297B   1326B   1664B   1696B   1705B   1826B   1851B
                2686B   3918B   3964B
3353  CHINTZ    2907B   3221B
2429  CHK1      2434B
2455  CHK2      2449B
2456  CHK3      2452B   2454B
2423  CHKBW     2182B   2318B
1774  CHKRNG    1645B   1660B
3759  CHLV      3020B   3308B   3515B
3762  CHLV1     3768B
3767  CHLV2     3763B   3765B
1256  CHPGT      471B
1266  CHPGT2    1264B
2261  CHRG1     2242C
2266  CHRG2     2223C   2238    2240
2310  CHRG3     2241C
2320  CHRG4     2237C   2243
2262  CHST1     2218C
2270  CHST2     2219C
2312  CHST3     2220C
2319  CHST4     2221C
4227  CHZ       4224B
3726  CNFILL    3698
2676  COMP      2608C   2679
 366  CON1       366B
 386  CON2       373B
 395  CON2A      427B
 421  CON2B      428B
 420  CON2C      398B    409B
 431  CON3       423B
6184  CONT1      432     516     561     922    1556    4249
6192  CONT10    4266
6193  CONT11    1514    3329    4002
6194  CONT12    3526
6185  CONT2      535     593    2945
6186  CONT3      927    4254
6187  CONT4      527     588     919    1636    1774    3867    4186
6188  CONT5     3344    3787    3828
6189  CONT6      603    2655    4324    4376
6190  CONT7      679    1355    1604    1652    3354    3984    4221
6191  CONT9     1559    4059
6137  COUNT     2519C   2531    2533C
2638  CP1       2607C   2643    2644C
2616  CP2       2609C
2672  CP3       2612C
1362  CPT2      1335
2584  CPT3      2614    2735
2587  CPT3A     2555C   2566C   2575C
1367  CPT4      1286
1366  CPT5      1317
4114  CRC       2594B   2732B   2789B
6111  CRC12     2573
6108  CRC16     2551    2728    2786
4132  CRCA      4138B
4126  CRCAA     4146B
4127  CRCAA1    4123B
6148  CRCAC     2801    2803    4147C
4138  CRCB      4136B
4144  CRCC      4180B
4145  CRCC2     4152B   4176B
4154  CRCD      4131B   4163B
4151  CRCDD     4140B
4163  CRCE      4161B
4178  CRCF      4129B
2513  CRCT       472B
6122  CRLF      4431    4467
 892  CRWB       784B    806B    850B    880B   1201B
2579  CRZR      2570B
6460  DATE      6458
      DCMC1        1    6468
6119  DFLT       886    1577    2127    2616
4352  DLAY       687B    937B   1168B   1288B   1301B   1341B   1609B   1695B   1703B   1830B
                1854B   2393B   2775B   2908B   3073B   3137B   3168B   3222B   3279B   3501B
4381  DLAY1     4400B
4368  DLAY2     4355B   4364B
4396  DLAY3     4390B
4388  DLAY4     4381B
4400  DLAY5     4394B
4365  DLAYLG    1454B   1573B   3114B   3929B   4064B
4360  DLAYLI
6230  DMP       6242B
6237  DMP1      6235C
6239  DMP2      6234C
3200  DOL2      3187B   3195
3277  DOL3      3261B
3369  DOL4      3366B
2393  DOLA      2390B
3291  DOLBB     3278B
6180  DON        475
6206  DONE       477
 348  DRSR       345B
 345  DRSR1      350C
```

```
3154  DSLOOP   3217B
3066  DSST      466B
3525  DSSTA    1340B   3126B
3070  DSSTLP   3313B
1716  DTPT     1651
6136  DUMMY     358     359     529     590    2156    2269    2373    2380    2386    2638
                2672    2793    2799    3421    3579    3962    6134    6402
1644  ECO3
1650  ECO2     1641
1652  ECO4     1650B
6101  ECP      1682
6139  ENDSAV   3714
2311  EOADD    2306C
2260  EODATA   2256C
6205  EQ        415
4440  ER11     4434
6164  ERAR      696     697     700     701     702     892     893     896     897     898
                2676    2677    2680    2681    2682
6177  ERF      4441
6207  ERMG      500C    636C    723C    911C    955C   1127C   1257C   1376C   1505C   1549C
               1809C   1893C   1907C   1921C   1934C   1949C   1962C   1983C   1998C   2012C
               2026C   2040C   2053C   2514C   2702C   2899C   3067C   3393C   4388    4430
6208  ERMG1     375
4430  ERR1     4429C
4443  ERR2     4469B
4450  ERR21    4445    4448B
4449  ERR21A
4458  ERR3     4433    4444    4451    4453B   4456C   4474    4479B
4457  ERR31    4438    4449
4463  ERR34    4459    4461B
4469  ERR4     4466B
4471  ERR5     4442B
5132  ERR6     5091    5092    5094    5103    5104    5106
3549  ERRINT   3637
4425  ERROR     407B    520B    525B    531B    539B    549B    565B    569B    576B    592B
                597B    601B    607B    611B    624B    683B    693B    703B    899B    926B
                931B    943B   1056B   1096B   1112B   1175B   1186B   1306B   1311B   1350B
               1359B   1474B   1518B   1538B   1563B   1567B   1571B   1608B   1629B   1640B
               1656B   1712B   1748B   1778B   1783B   1787B   1839B   1847B   2167B   2178B
               2195B   2285B   2298B   2331B   2433B   2442B   2458B   2666B   2683B   2768B
               2805B   2975B   3118B   3144B   3204B   3215B   3283B   3287B   3302B   3333B
               3348B   3358B   3530B   3589B   3596B   3766B   3848B   3856B   3860B   3864B
               3873B   3933B   3936B   4006B   4063B   4190B   4225B   4253B   4258B   4270B
               4328B   6449B
3594  ERUPT    3553
4600  ERZ1     4591    4592    4593    4610    4611    4612
2487  FDTA     2130B   2259B
2502  FDTA1    2503B
2495  FDTALP   2497B
 760  FILL      762B
6141  FLAG     2097C   2106C   2164    2231C   2981C   2984    3021    3023C
4041  FLN       510B    558B    585B    649B    731B    916B    960B   1133B   1214B   1262B
               1380B   1508B   1552B   1813B   2112B   2249B   2526B   2708B   2808B   2904B
               3070B   3414B   3487B   4057B
4044  FLN2     4047B
4050  FLN3     4045B
6454  FREV     6433
6114  FRST      339     351     705     708C
3744  FTBL     3439B   3469B   3474B
6125  FW-REV   6443C   6444    6446
4055  GENITZ    504B    554B    639B    726B    914B    957B   1130B   1260B   1384B   1513B
               1678B   1811B   2128B   2257B   2516B   2706B   2903B   3072B   3409B   6425B
3041  GNBPRG   2923
4604  GNZZ     4596    4597    4598
 993  GRB1      987B
6179  HRTZ      354C    356     4353    4361    4366
1528  ICON1    1525B
1531  ICON2    1523C
1508  ICON3    1540B
2963  IDLP     2965B
6204  IDMSG     410
3841  IHD1     3707    3875B   6271    6279    6287    6295    6303    6311    6319    6327
               6335    6343    6351    6359    6367    6375    6383    6391
3856  IHDW1    3854B
3861  IHDW2    3858B
3865  IHDW3    3862B   3869
3874  IHDW4    3866B   3871B
3849  IHDW5    3846B
1871  ILC1     1868B
1865  ILCTST   1105B   1843B   2174B   2294B   3115B   3141B   3201B   3280B
6417  ILV      2918C   2993C   3177C   3241C   3442    3455    3683    3762    3861    3985
               4000    4019
6118  IMASK     388C    420    3435
6155  IMGA      507C    518     563     574    4122C   4145
3881  IMON     3544    3614    3891B
3888  IMON2    3883B
3890  IMON4
4265  INBYTE   1052B   1173B   1873B   1877B   2190B   2326B
1504  INCONT    463B
3617  INLLST   3617    3631
2903  INLOOP   3037B
6409  INMB     2996C   3244C   3472    3688    3764    3853    3888C
1465  INS1     1459C
1387  INS2     1385B
4852  INS2A
5148  INS4     5181    5182    5183    5191
5163  INS5     5150    5151    5152    5158
5179  INS6     4854    4855    4856    4861
1380  INSLUP   1477B
2898  INST1     465B
2982  INST2    3031B
3036  INST3    3022B
2988  INST4    2985B
4498  INSTR1   1397    1398
4850  INSTR2   1398    1404    1405
5008  INSTR3   1405    1411    1412
5141  INSTR4   1412    1417    1418
5156  INSTR5   1418    1423    1424
5172  INSTR6   1424    1429    1430
1375  INSTT     462B
3609  INTBL    3629
3628  INTITZ    502B   2906B   2982B   3175B   3239B   3410B
3098  INTLCT   2911
```

```
 510  NXCH      547B
2526  NXCHC     2530B   2580B
2531  NXCHC1    2528B
2537  NXCHC2    2535B
2524  NXCHC4    2527B
 976  NXLCT      981B
 985  NXLCT1     989B
 968  NXLCTA     963B
2112  NXRCV     2118B   2198B
 516  NXRG       545B
2282  NXT1      2280B
2332  NXT2      2330B
 561  NXTCCB     572B
2249  NXWRT     2255B   2333B
4482  OPT       4436    4439
2701  OTPT       473B
2808  OTPT1     2812B
2708  OTPT2     2712B
2795  OTPT3     2780C
2713  OTPT4     2710B
2734  OTPT5     2810B
2770  OTPT7     2767B
3343  OUTLCT    1051B   1172B   1872B   1876B   2189B   2325B   3113B   3136B   3163B   3172B
                3209B   3251B   3269B   3294B
 494  PASMSG     487
6116  PASS       343C    488C    489
4291  PKR1      4297B
4287  PKR1N     1081B
6120  PNOP      1270
6424  PREV       707B
2807  PRFR1     2802B
3618  PRIEND    3882
3399  PRILP     3520B
6420  PRIST     3616    3618    3619    3692
1244  PROG1     1279
 487  PSPT       484B
3376  PVLU1     3372C
 834  Q1         823C    829
 835  Q2         819C    853
 845  Q3         824C
 846  Q4         820C
 865  Q5         816C    858
 875  Q6         817C
5442  R55       5433    5434    5435
5406  R56       5357    5358    5362    5369    5370    5371    5382    5383    5387    5394
                5395    5396
5348  R57       5217    5218    5219
5360  R58       5228    5229    5231
5373  R59       5238    5239    5240
5385  R60       5249    5250    5252
5398  R61       5259    5260    5261
5293  R67       5287    5288    5289
5336  R68       5297    5298    5300
5340  R69       5313    5314    5315
5310  R70       5282    5283    5284
5324  R71       5318    5319    5320
3422  RAD       3402C   3427C   3428C
6153  RAMAD      753C    895
6154  RAMAD2
6150  RANGE      646C    665     673     920C    1576C   1577
6151  RANGE1    2523C   2616    2676
6152  RANGEW     646C    696     752C    758     786C    826C    860C    886     892     1200C
                1463    1469    1672C
1057  RBYTE1    1055B
1047  RBYTES    1059B
1892  RC0        468B
1906  RC1       1896B
2144  RC11      2122C
1920  RC2       1909B
1933  RC3       1923B
1948  RC4       1936B
1961  RC5       1951B
 582  RCCBS      559B
2131  RCDATA    2123C
5468  RCDT
5466  RCDTC1    2154    2155    2268
2200  RCEND     2115B
2380  RCLCT     2367C
5734  RCRT1     2155    2544    2545
2349  RCV       2151B
2133  RCV1      2079C
2156  RCV10     2091C
2196  RCV11     2194B
2119  RCV12     2116B
2107  RCV13     2099B   2104B
2105  RCV14     2100B
2372  RCV1M     2370B
2145  RCV2      2080C
2373  RCV2M     2362C
2157  RCV3      2081C
2386  RCV3M     2363C
2183  RCV4      2082C
2387  RCV4M     2384C
2132  RCV5      2095C   2102C
2143  RCV6      2096C
2153  RCV7      2085C   2098    2101
2184  RCV8      2094C   2103C
2158  RCV9      2088C   2169
2074  RCVTST    1895B   1908B   1922B   1935B   1950B   1963B
2374  RCWK      2365C
2375  RCWK1     2076C   2351C   2414C
3952  RDATA      777B    799B    843B    873B   1065B   1192B   1462B   1528B   1618B   1733B
                2308B   2792B   3373B   6236B   6426B
3962  RDTA1     3954C
3963  RDTA2     3957C
 713  READ       663C    678
 343  RESTRT
 549  RGE        541B
 499  RGNT       457B
 544  RGNT1      550B
5736  RMCRT     5895    5896    5898
5804  RMCRT1    5820    5821    5822
5890  RMCRT2    5883    5884    5885
```

```
5788  RMCRT3   5749   5750   5752
5848  RMCRT4   5767   5768   5770   5811   5812   5814   5831   5832   5833
5756  RMCRT5   5784   5785   5786
5825  RMCRT7   5801   5802   5806   5845   5846   5850
5871  RMCRT8   5865   5866   5868
6172  RMINST    675   6157
4536  RMOP
5908  RMOPT
5928  RMOPT1   5966   5967   5969
5945  RMOPT2   5939   5940   5942
5918  RMOPTA   5986   5987   5989
2958  RNFC     2947C  2964
6121  RNG      1269C  1270
 588  RNXR      615B
5906  ROD1     2545   2718   2719
1813  RPSIO    1857B
3364  RPVLU    1162B  1302B  1343B  1708B  2446B  3211B  3298B
 351  RST       346B   376B
 357  RST1      352B
3432  RT1      3415B
3493  RT10     3502B
3471  RT2      3462C
3476  RT3      3463C
3452  RT5      3445B
3434  RT6      3448B
3455  RT7      3460B
3447  RT7A     3436B
3460  RT8      3458B
6253  RTB       412C   413    651C   652C   653C   654C   668    696    778    800
                844    874    886    892   1066   1091   1193   1449   1577   1619
               1627   1734   1745   1795   1821   2127   2152   2309   2427   2500
               2934   3005   3026   6232   6237   6251   6427   6434   6441
4410  RTC      2953B  3009B  3183B  3257B  3506B
3605  RTCFLG   3603C  3664C
3603  RTCHDL   3562   3610
6173  RTCHN    3327C
3558  RTCSAV   3657
3562  RTCSV1   3609
5200  RTD      5303   5304   5305   5329   5330   5331   5450   5451   5452
5204  RTD1     5196   5197   5198   5437   5438   5439   5444
5269  RTD2     5428   5429   5430
4019  SAF4
4020  SAF5
6257  SAF9     6254
4483  SAFF     4473
6165  SAV1     3628C  3723   3744C  3754   3759C  3769   3976C  3981   4041C  4050
               4218C  4227   4244C  4260   4287C  4298   4307C  4335   4352C  4360C
               4365C  4402
6166  SAV2     1865C  1880   3777C  3784   3906C  3937   3952C
6168  SAV3     2349C  2396   2410C  2487C  2504   3525C  3531   4055C  4065
6169  SAV4     2074C  2200   2215C  2335   4114C  4148
6171  SAV5     4425C  4443   4455   4471
6170  SAV6     4185C  4192
6167  SAVN     3796C  3825
6158  SBCCB     642C   696
6159  SBCCP
6157  SBCK      641
 554  SCCBTS    533B
6146  SCFLG    3246C  3666C  3865
3235  SCIT1    3232
3221  SCIT1A   3158B
3238  SCIT2    3233B  3271
3266  SCIT3    3255   3270
3317  SCNPRG   3078
3094  SCTEST   3091   3155   3225
3110  SCTST1   3092B
3338  SCTSVA   3194
3906  SDATA     769B   789B   832B   863B  1145B  1153B  1271B  1278B  1396B  1403B
               1410B  1416B  1422B  1428B  1437B  1680B  2141B  2372B  2543B  2717B
               2922B  3077B
6248  SDB       757    770    790    833    864    892    978C   988C   994C   996C
                997C  1032   1043   1046   1075   1082   1083   1090   1138   1146
               1270   1272   1589C  1590C  1591C  1592C  1593C  1594C  1596   1818
               2142   2265   2426   2488   2490C  2520   2552   2604   2676   2691
               2727   2739   2744   2749   2754   2772   2785   3399   6246   6424C
               6452
6174  SEC       449C   482    514    515C  4345C
 886  SET       755B   796B   840B   870B  1673B
4017  SET1     4026B
4023  SET2     4021B
1793  SET4     1600B  1644B  1659B  1692B
1794  SETA     1799B
3777  SETLCT   1161B  1285B  1316B  1334B  1445B  1688B  1817B  2379B  2613B  2734B
               2910B  3029B  3090B  3154B  3193B  3224B  3231B
3602  SETNEW   3604B
4474  SETNR    4447B
3999  SETV     3485B
4008  SETV1    4001B
4000  SETV2    4010B
4015  SETVEC   2943B  2997B  3242B  3481B
3033  SFLG     3030
1130  SIT1
1202  SIT10
1192  SIT11    1189B
6441  SIT12    6438B
1169  SIT2
1145  SIT4     1142B
1176  SIT5     1174B
1171  SIT6     1177B
1186  SIT7     1165B
1141  SIT9     1216B
1126  SITEST    461B
3392  SITST     467B
3750  SLVLP    3753B
3796  SNDLCT   1042B  1595B
6123  SPACE    4432   4476
4480  SPACE1   4462
3917  SPRG2    3911C
3924  SPRG3    3968B
3916  SPRG5    3909C
3929  SPRG7    3924B
3931  SPRG8    3928B
4500  ST       4691   4692   4693
```

```
6178  STAT      4191C   4399C
1808  STIORC     464B
3619  STKPTR    3615    3881    3886C
 446  STLOOP     340B
 335  STOP       344C    447C    476B    478B    492C    493B
 339  STRT       341     491B   6468
1355  STRTIO    1167B   1289B   1318B   1451B   1694B   1702B   2391B   2650B   2758B
3649  SVEC      3650B
4481  SWT       4435C   4452    4460    4475C
3984  TALL      3980B
1498  TALU      1438
6143  TEMP      1529    1535    1565    1776    3374    3379    3381C   4117C   4125C   4139
                6437C   6439
6144  TEMP1      764C    882    1383C   3129C   3169    4118C
4689  TEMPA     4865    4866    4868
6145  TEMPST    3135C   3250
2636  TESB      2645B
2594  TESC      2558B   2568B   2577B   2601
2597  TESCA     2606B
2607  TESCC     2603B
2667  TESD      2662B
2661  TESE      2654B
2650  TESF      2663
2653  TESG      2674B
2655  TESH      2664B
2674  TESJ      2668B
 750  TESM       739B    741B    743B    745B
 882  TESM1      854B
4307  TEST      2762B   3925B
4319  TESTZ     4327B
4330  TESTZ1    4320B
6140  TIMEIN
4695  TMP1      4685    4686    4687
6399  TOPSAV    3654    3721C   3722C
6142  TPR        396C    414     418C    419     659C    672     694     765C    767C    768C
                 776C    788C    798C    828C    842C    862C    872C   1141C   1166    1188
                1212    1267C   1382C   1475    1524C   1534    1580   1611C   1615    1632
                1729C   1739    1874C   1879    2369C   2378    2448   2920C   2927    3075C
                3083    3365C   3382    4413C   4415
1982  TRAN1      469B
1997  TRAN2     1986B
2011  TRAN3     2000B
2025  TRAN4     2014B
2039  TRAN5     2028B
2052  TRAN6     2042B
5194  TRANM1    1430    2267    2268
3976  TRNON     2960B   3493B   3497B
3587  TRPERR
6156  TSA2      3673
6209  TSTA       501
6210  TSTB       637
6211  TSTC       724
1119  TSTCHN     962C   1060    1114
6212  TSTD       912
6213  TSTE       956
6215  TSTF      1258
6216  TSTG      1377
6217  TSTH      1506
6218  TSTJ      1550
6219  TSTK      1810
6220  TSTL1     1894
6221  TSTM1     1984
6222  TSTN      2515
6223  TSTP      2703
6224  TSTR      2900
6225  TSTS      3068
6214  TSTSI     1128
6226  TSTT      3394
1236  TSVLU     1169
6175  TTOT       450C   4339    4340C   4344C
4339  UPTM      4333B   4371B   4414B
4346  UPTM1     4343B
6243  V$LB      6467
5998  VALTST    1681    1682    2719
6449  WFW       6451B
2410  WRT       2264B
2361  WRT1      2415B
3270  WRUPT     3272B
5709  XM55      5701    5702    5704
5674  XM56      5624    5625    5626    5637    5638    5642    5649    5650    5651    5662
                5663    5667
5615  XM57      5491    5492    5493
5628  XM58      5502    5503    5505
5640  XM59      5512    5513    5514
5653  XM60      5523    5524    5526
5665  XM61      5533    5534    5535
5567  XM67      5561    5562    5563
5603  XM68      5571    5572    5576
5608  XM69      5589    5590    5591
5580  XM70      5556    5557    5558
5574  XM71      5594    5595    5596
5474  XMD       5579    5580    5581    5716    5717    5718    5720
5478  XMD1      5470    5471    5472    5705    5706    5707
5543  XMD2      5696    5697    5699
5193  XMDT
2215  XTST      1985B   1999B   2013B   2027B   2041B   2054B
2256  XTST1     2253B
2230  XTST2     2228B
2241  XTST3     2239B
 320  ZERO       334    6245    6247    6250    6252    6255    6257
      ZHCOMM     326
      ZHIAFB     323    3667C   3668C   3669C   3671C
      ZHISAZ     323    3638    3648    3653    4022C   4369C
      ZHNTSA     321     362C   3652C
      ZHPFR      326    3646    4308    4368
      ZHRTCC     324    4315C   4321    4332    4370C   4393    4396    4412C
      ZHRTCI     325    4310C   4373C   4411C
      ZHRTCL     325    4312C   4375C   4417C
      ZHTH15     321     364C
      ZHTH17     322
      ZHTSA      322     361    3651    3674C
      ZHWDTC     324
      ZV$C       696B    892B   2676B
```

```
          ZV$CO      704B     900B    2685B
          ZV$ER      375B    4430B
          ZV$F       386B     886B    1138B    1270B    1577B    2127B    2616B    3679B    3683B    3688B
                    3692B    3698B
          ZV$HM      921B
          ZV$HR      321     2600
          ZV$ID      356B
          ZV$IH      358B
          ZV$PCH     486B
          ZV$QC      355B     357B
          ZV$RD      322      348B     921B
          ZV$T       415B     475B     477B     501B     637B     724B     912B     956B    1128B    1258B
                    1377B    1506B    1550B    1810B    1894B    1984B    2515B    2703B    2900B    3068B
                    3394B    4462B
          ZV$TC      410B     475B     477B     487B     501B     637B     724B     912B     956B    1128B
                    1258B    1377B    1506B    1550B    1810B    1894B    1984B    2515B    2703B    2900B
                    3068B    3394B    6433B
          ZV$TD      413B    6439B    6444B
          ZV$TH      419B    4457B
          ZV$THZ    4458B
  715 LABELS
 4247 REFERENCES
 6468 RECORDS
    1 U FLAGS
    0 M FLAGS
   38 N FLAGS
 6 CROSS REF VERSION L - 24 SEPT, 1976
 RS LINKER VERSION 5.00 05/10/78  2055.8 EDT WED
 LINK MAP FOR DCMC1
  START          0100
  LOW            0000
  HIGH           2BA9
  CURRENT        1F31
 *LOC DEFS
   ZHCOMM        0000
  *DCMC1         0000
   ZHPFR         0000
   ZHTSA         0002
   ZHNTSA        0010
   ZHRTC1        0014
   ZHRTCC        0015
   ZHRTCL        0016
   ZHWDTC        0017
   ZHMERC        001F
   ZH1AFB        0020
   ZHTH29        0063
   ZHTH28        0064
   ZHTH27        0065
   ZHTH26        0066
   ZHTH25        0067
   ZHTH24        0068
   ZHTH23        0069
   ZHTH22        006A
   ZHTH21        006B
   ZHTH20        006C
   ZHTH19        006D
   ZHTH18        006E
   ZHTH17        006F
   ZHMEMP        006F
   ZHTH16        0070
   ZHLERR        0070
   ZHTH15        0071
   ZHNRES        0071
   ZHTH14        0072
   ZHPMEM        0072
   ZHTH13        0073
   ZHP-OP        0073
   ZHTH12        0074
   ZHTH11        0075
   ZHTH10        0076
   ZHTH9         0077
   ZHTH8         0078
   ZHTH7         0079
   ZHTH6         007A
   ZHOVFL        007A
   ZHTH5         007B
   ZHOP-N        007B
   ZHTH4         007C
   ZHTH3         007D
   ZHSC-N        007D
   ZHTH2         007E
   ZHTRC         007E
   ZHTH1         007F
   ZHMCL         007F
   ZH1SAZ        0080
   ZH1VBS        0080
   ZHTVBS        0080
  *ZV$TH         1664
   ZV$TD         1699
   ZV$TH         1664
   ZV$THZ        168C
  *ZV$IH         16B4
   ZV$ID         16B9
   ZV$IH         16B4
   ZV$IAD        16BE
   ZV$---2       16D6
   ZV$---3       16E8
  *ZV$ER         174D   REV. 5.0
   ZV$ER         174D
   ZV$TA         1779
   ZV$---0       1760
  *ZV$F          17BD
   ZV$F          17BD
  *ZV$T          17CB   REV. 5.0
   ZV$QC         17E8
   ZV$TC         17D4
   ZV$T          17CB
   ZV$Q          17DD
  *ZV$PCH        17FC
   ZV$PCH        17FC
  *ZV$C          18FE   REV. 5
   ZV$C          18FE
   ZV$CO         1921
```

```
*ZV$GP          1932
 ZV$GP          1932
 ZV$--4         1952
*ZV$HA          195E
 ZV$HA          195E
 ZV$HZ          1968
 ZV$HS          1963
*ZV$HD          1997
 ZV$HD          1997
*ZV$IA          19C9   REV. 7
 ZV$IA          19CC
 ZV$ABF         1A7D
 ZV$AKG         1A7B
 ZV$--1         1A38
 ZV$IAV         19CA
*ZV$BKK         1A88
 ZV$BKK         1A88
*ZV$RD          1AA2   REV. 7
 ZV$RD          1AA2
 ZV$HM          1B18
 ZV$HK          1AD1
 ZV$SV1         1C77
 ZV$SV3         1C97
 ZV$AF          1AB3
 ZV$TTY         1AB5
 ZV$SV2         1C87
 ZV$OTP         1B49
 ZV$BKF         1ACA
 ZV$TID         1AB4
 ZV$CF2         1ABE
 ZV$TK          1ABA
 ZV$RAR         1ABB
 ZV$ST1         1ABF
 ZV$RCC         1AC0
 ZV$BUD         1AB6
 ZV$OLB         1AC2
 ZV$RCB         1AC3
 ZV$NSR         1AC7
 ZV$STR         1AC5
 ZV$BKS         1AC9
 ZV$IZ          1ADC
 ZV$LR          1ACE
 ZV$DAT         1AB1
 ZV$HRU         1ACB
 ZV$HRL         1ACC
 ZV$LRU         1ACD
 ZV$LRL         1ACE
 ZV$HBD         1ACF
 ZV$CF1         1ABD
 ZV$--5         1AD4
 ZV$RMD         1AB2
 ZV$MCP         1AD0
 HIBAUD         1ACF
 ZV$RAW         1ABC
 ZV$RDT         1CD3
 ZV$CTL         1AB9
 ZV$B1          1BF4
 ZV$TST         1D29
 ZV$MDC         1CFD
 ZV$R99         1EFB
 ZV$ISA         1AD7
 ZV$UIH         1AD2
 ZV$ZRO         1B56
 ZV$BSH         1B58
 ZV$CPU         1AB8
 ZV$R50         1B36
 ZV$R60         1B41
 ZV$RT          1E38
 ZV$ALL         1AB7
*MLCHPG         1F00   T+V
 MLCHPG         1F00
 ENDCHP         1F31
*UNLINK MODULE(S)
 ZV$QC
 ZV$ID
 ZV$TC
 ZV$TD
 ZV$CO
 ZV$TH
```