```
000001                          TITLE MLCS1, 'REV F' MLCP TEST (SAF)
000002                  *
000003                  *
000004                  *
000005                  *
000006                  * DESCRIPTION:
000007                  *
000008                  *    THIS T&V PROGRAM VERIFIES PROPER OPERATION OF THE MULTILINE
000009                  *    COMMUNICATION PROCESSOR (MLCP) FOR THE SERIES 60, LEVEL
```

```
000010                  *    6/3X, 6/4X, AND 6/5X.
000011                  *
000012                  *    THE SUBSYSTEM ITEMS SUPPORTED BY THIS PROGRAM ARE:
000013                  *
000014                  *       MLC9101 MULTILINE COMMUNICATION PROCESSOR
000015                  *       MLC9102 MULTILINE COMMUNICATION PROCESSOR
000016                  *       MLC9103 MULTILINE COMMUNICATION PROCESSOR
000017                  *
000018                  *
000019                  * REVISION HISTORY:
000020                  *
000021                  *       REV     DATE
000022                  *        A      JUNE 76     ORIGINAL RELEASE
000023                  *        B      SEPT 76
000024                  *        C      JAN  77
000025                  *        D      APR  77
000026                  *        E      AUG  77
000027                  *        F      JUNE 78
000028                  *
000029                  *
000030                  *
000031                  ********************************************************************************
000032                  *    THIS  DOCUMENT  AND  THE  INFORMATION  CONTAINED
000033                  *    THEREIN IS CONFIDENTIAL AND PROPRIETARY TO AND THE
000034                  *    EXCLUSIVE  PROPERTY  OF  HONEYWELL INFORMATION
000035                  *    SYSTEMS INC.  IT  IS MADE AVAILABLE ONLY TO HONEY-
000036                  *    WELL AUTHORIZED RECIPIENTS FOR THEIR USE SOLELY IN
000037                  *    THE  MAINTENANCE  AND  OPERATION  OF  HONEYWELL
000038                  *    PRODUCTS.  THIS  DOCUMENT  AND INFORMATION MUST BE
000039                  *    MAINTAINED IN STRICTEST CONFIDENCE;  IT  MUST  NOT
000040                  *    BE  REPRODUCED  IN  WHOLE OR IN PART; AND IT SHALL
000041                  *    NOT BE DISCLOSED TO ANY OTHER  PARTY  WITHOUT  THE
000042                  *    PRIOR WRITTEN CONSENT OF HONEYWELL.
000043                  *
000044                  *
```

```
000045                          / PROGRAM PREPARATION:
000046                          *
000047                          *    THE ROOT SOURCE OF THIS PROGRAM, AFTER THE ADDITION OF APPROPRIATE
000048                          *    TITLE AND END STATEMENTS, WAS PROCESSED BY THE HOST RESIDENT ASSEMBLER
000049                          *    TO CREATE EITHER SHORT OR LONG ADDRESS FORM (SAF OR LAF) OBJECT TEXT
000050                          *    AND LISTING. THE OBJECT TEXT WAS FURTHER PROCESSED BY THE HOST RESIDENT
000051                          *    LINKER USING THE APPROPRIATE CONSOLE ZVSLIB LIBRARY TO CREATE A PUNCH
000052                          *    SEGMENT CONTAINING AN EXECUTABLE MODULE. THE ASSEMBLY LISTING WAS
000053                          *    AUGMENTED WITH CROSS REFERENCE DATA PLUS THE LOAD MAP FROM THE LINKER
000054                          *    TO CREATE A LIST SEGMENT.
000055                          *
000056                          *                ROOT          SAF           LAF
000057                          *                ----          ---           ---
000058                          *        NAME    MLCX1         MLCS1         MLCL1
000059                          *        DOCUMENT 60133776-006  60129825-006  60133777-006
000060                          *
000061                          *
000062                          * PROGRAM DISTRIBUTION:
000063                          *
000064                          *    THE ELEMENTARY ITEMS SUBMITTED TO THE I&V PROGRAM DISTRIBUTION CENTER
000065                          *    WERE THE EXECUTABLE PUNCHED CARD DECKS OF MLCS1 AND MLCL1, AND MAGNETIC
000066                          *    TAPE IMAGES OF THE AUGMENTED LISTINGS.
000067                          *
000068                          *    REPRODUCTIONS OF THE EXECUTABLE CARD DECKS MAY BE AS DUPLICATE CARD DECKS
000069                          *    OR AS MEMBER  OF A MULTIPLE MEMBER FILE. IN THE MOST FREQUENT CASE
000070                          *    IT WILL BE FOUND AS A MEMBER "SK" (SAF) OR "LK" (LAF) WITHIN FILE
000071                          *    PROGFILE OF A DISKETTE VOLUME ENTITLED DIAGS.
000072                          *
000073                          *    DISTRIBUTION OF THE LISTINGS, WHICH SHOULD BE AVAILABLE IF ANY COMPLEX
000074                          *    MAINTENANCE OR REPAIR IS TO BE PERFORMED, IS NORMALLY MADE AS A
000075                          *    PRINTED COPY.
000076                          *
000077                          *
000078                          * ROUTINE DEMONSTRATION:
000079                          *
000080                          *    A MINIMUM SATISFACTORY TEST FOR NORMAL OPERATION MAY BE OBTAINED
000081                          *    BY ENTERING THE CHANNEL NUMBER TO BE TESTED AND COMPLETING ONE PASS.
000082                          *
000083                          *
000084                          * MAIN MEMORY REQUIREMENT:
000085                          *
000086                          *    THIS PROGRAM REQUIRES 16K WORDS OF MAIN MEMORY AND WILL USE
000087                          *    ALL OF AVAILABLE MEMORY THROUGH 64K WORDS.
000088                          *
000089                          ************************************************************************
000090                          *
000091                          *
000092                          *
000093                          *
000094                          * TEST PROCEDURE AND DESCRIPTION
000095                          *
000096                          *   THE PROGRAM AUTOMATICALLY DETERMINES WHAT CHANNELS ARE ACTIVE
000097                          *   ON THE MLCP TO BE TESTED. IT PROCEEDS TO EXCERCISE ALL MOTHER
000098                          *   BOARD FUNCTIONS THAT DO NOT REQUIRE LINE ADAPTERS. THE TEST INCLUDES
000099                          *   MLCP MEMORY, INSTRUCTION SET, AND DATA XFER TESTS. PSUEDO RANDOM
000100                          *   DATA IS TRANSFERED VIA ALL CHANNELS. THE MLCP IS MADE TO TRANSFER
000101                          *   DATA TO ALL AVAILABE MEMORY ABOVE THE PROGRAM. MLCP INTERRUPTS ARE
000102                          *   MADE ON ALL ACTIVE CHANNELS AT MULTIPLE LEVELS.
000103                          *
000104                          *   AN ACTIVE CHANNEL IS ONE WHICH EITHER HAS A LINE ADAPTER PRESENT
000105                          *   OR HAS ITS LINE ADAPTER-HERE SIGNAL TIED TO GROUND. THE  TEST
000106                          *   PROGRAM CAN ONLY TEST ACTIVE CHANNELS.
000107                          *
000108                          *
000109                          * OPERATING INSTRUCTIONS
000110                          *
000111                          *     LOAD AND START (OR RESTART) THE PROGRAM. THE PROGRAM IDENTIFICATION WILL
000112                          *     BE DISPLAYED ON THE CONSOLE.  THE INITIAL START WILL ALSO DISPLAY:
000113                          *
000114                          *       THE ZVSLIB REVISION NUMBER
000115                          *       THE ADDRESS FORM (SAF OR LAF)
000116                          *       I/O  EQUIPMENT DETECTED IN THE SYSTEM
000117                          *       MEMORY SIZE
000118                          *
000119                          *     THIS DISPLAY MUST BE VERIFIED BY THE OPERATOR.   THIS DISPLAY IS OMITTED
000120                          *     ON RESTARTS.
000121                          *
000122                          *     THE CONSOLE SEARCH RULES ARE:  FIND THE CONSOLE WITH THE LOWEST  CHANNEL
000123                          *     NUMBER CONNECTED THRU AN  MDC  CONTROLLER.  IF THERE IS NO CONSOLE ON AN
000124                          *     MDC, THEN SEARCH FOR A TERMINAL WITH THE HIGHEST CHANNEL NUMBER ASSIGNED
000125                          *     TO AN ACLA ADAPTER ON AN MLC CONTROLLER.  IF NO ASYNC ADAPTER  IS FOUND,
000126                          *     THEN GO TO THE FULL CONTROL PANEL.
000127                          *
000128                          *     THERE ARE THREE CONSOLE CHANNEL OPTIONS DETERMINED BY THE VALUE  OF  LO-
000129                          *     CATION "ZV$TTY".
000130                          *
000131                          *     IF ZV$TTY EQUALS (0000), SEARCH FOR A CONSOLE.
000132                          *     IF ZV$TTY EQUALS (FFFF), ASSUME THERE IS NO CONSOLE.
000133                          *     IF ZV$TTY EQUALS NEITHER (0000), NOR (FFFF), THEN IT IS THE CONSOLE CHAN-
000134                          *     NEL NUMBER.   NOTE:  DEFAULT IS TO SEARCH FOR A CONSOLE.
000135                          *
000136                          *     ALL CONSOLE I/O IS EVEN PARITY.  IF CONSOLE IS ON MLC,  IT MUST BE ASYNC
000137                          *     AND THE BAUD RATE SET AT 1200 TO MATCH THE PROGRAM SUPPLIED RATE.  IF IT
000138                          *     IS NECESSARY TO CHANGE THE PROGRAM BAUD RATE,   THEN THE  NEW  BAUD RATE
000139                          *     CODE SHOULD BE PUT INTO LOCATION "ZV$BUD" IN HEX. THE TERMINAL BAUD RATE
000140                          *     MUST BE  SET  TO MATCH THIS NEW BAUD RATE.  THE CORRECT HEX VALUE MAY BE
000141                          *     OBTAINED FROM THE FOLLOWING TABLE.
000142                          *
000143                          *-----------------------------------------*
000144                          *                                         *
000145                          *        BAUD  RATE  TABLE                *
000146                          *                                         *
000147                          *-----------------------------------------*
000148                          * ACLA I.D.E(2118)(2110) E(2108)
000149                          * BAUD-RATE
000150                          *    50    E   0 EE   1
000151                          *    75    E   1  EE   2
000152                          *   110    E   2 EE   3
000153                          *   134    E   3 EE   4
000154                          *   150    E   4 EE   5
000155                          *   200    E   5 EE   ---
000156                          *   300    E   6 EE   6
000157                          *   600    E   7 EE   7
```

```
000158                          *   900   E  --- EE   8
000159                          *  1050   E   8 EE  ---
000160                          *  1200   E   9 EE   9
000161                          *  1800   E  10 (A) EE  10 (A)
000162                          *  2000   E  11 (B) EE  ---
000163                          *  2400   E  12 (C)  E  11 (B)
000164                          *  3600   E  --- EE  12 (C)
000165                          *  4800   E  13 (D) EE  13 (D)
000166                          *  7200   E  --- EE  14 (E)
000167                          *  9600   E  14 (E) EE  15 (F)
000168                          * 19200   E  15 (F) EE  ---
000169                          *
000170                          *     TO MAKE ANY OF THE ABOVE CHANGES,  LOAD AND HALT THE PROGRAM BEFORE EX-
000171                          *     ECUTION.  INSERT CHANGE THEN EXECUTE.  MEMORY LOCATIONS OF "ZV$TTY" AND
000172                          *     "ZV$BUD" MAY BE FOUND IN MAP AT END OF LISTING.
000173                          *     CONSULT LEVEL-6 T&V MANUAL "AW94" FOR DETAILS ON HOW TO LOAD THE TESTS.
000174                          *
000175                          *     THE FOLLOWING IS A TYPICAL RESULT OF LOADING  AND  STARTING TO  RUN THE
000176                          *     PROGRAM.
000177                          *
000178                          *     MLCP TEST <PGM NAME> <PGM DATE> <PGM REV>
000179                          *     ZV$LIB REV. 7.0
000180                          *     ZV$AF= 1   <2>
000181                          *     WDT
000182                          *     CHAN  DEVC  ID
000183                          *     0400  DSKT  2010
000184                          *     0480  DSKT  2010
000185                          *     0580  CDR   2008
000186                          *     1200  DISC  2330
000187                          *     1280  DISC  2330
000188                          *     1300  LPT   2000
000189                          *     1380  CONS  2019
000190                          *     FD00  HDLA  2140
000191                          *     MEMORY LOW  00002B2D
000192                          *     MEMORY HIGH 00003FFF  16K
000193                          *
000194                          *   FOR 6/3X SYSTEMS ONLY THE PROGRAM WILL NEXT ASK:
000195                          *
000196                          *              PWR FREQ (HZ)?:
000197                          *
000198                          *    THE OPERATOR MUST TYPE IN THE FREQUENCY OF THE CLOCK
000199                          *    USED TO DRIVE THE REAL TIME CLOCK ON HIS SYSTEM. THE
000200                          *    PROGRAM NEXT DISPLAYS:
000201                          *
000202                          *
000203                          *              CHANNEL?:
000204                          *
000205                          *    THE OPERATOR MUST RESPOND WITH THE LOWEST ACTIVE ADDRESS ON
000206                          *    THE MLCP. ENTER 4 HEX DIGITS FOLLOWED BY A CARRIAGE RETURN.
000207                          *    THE MLCP ADDRESS WILL NORMALLY BE DISPLAYED AS AN ITEM IN THE
000208                          *    CONFIGURATION PRINTOUT.
000209                          *
000210                          *
000211                          *    THE PROGRAM NEXT DISPLAYS:
000212                          *
000213                          *              NEXT?:
000214                          *
000215                          *
000216                          *    VALID RESPONCES ARE:
000217                          *
000218                          *         A - RUN ALL TESTS.
000219                          *         T - SET FLAG TO PRINT ALL TEST LABLES
000220                          *         Q - QUICK MODE. RUN ONE PASS AND RETURN TO "NEXT".
000221                          *             Q TEST DOES AN ABREVIATED VERSION OF THE RANGE TEST BUT
000222                          *             RUNS ALL OTHER TESTS NORMALLY. IT USES ONLY 16 PATTERNS
000223                          *             FOR THE RANGE TEST RATHER THAN 65536.
000224                          *
000225                          *      NOTE - "T" AND "Q" RETURN TO NEXT. THEY DO
000226                          *             NOT TAKE EFFECT UNTIL MODE "A" IS
000227                          *             ENTERED.
000228                          *
000229                          *         I - INITIALIZE. CLEAR T, Q, + R FLAGS AND RESTART PROGRAM.
000230                          *         F - READ AND PRINT FIRMWARE OPTION TYPE AND REV.
000231                          *         R - RUN ONLY INTERRUPT AND DATA SET SCAN TESTS
000232                          *
000233                          *    THE PROGRAM WILL THEN RUN ABOUT 1 MINUTE AND TYPE:
000234                          *
000235                          *              MLCP OPT/FW REV  XX
000236                          *
000237                          *    THIS IS A PRINTOUT OF THE CURRENT FIRMWARE OPTION TYPE.
000238                          *    AND REVISION NUMBER. CURRENT FW OPTIONS ARE:
000239                          *
000240                          *         1 - COMM BOOT STRAP OPTION SUPPORT
000241                          *         2 - BIT STREAM MODE.
000242                          *
000243                          *    THE ABOVE  PRINTOUT IS GIVEN ON THE FIRST PASS ONLY.
000244                          *    VALUE "XX" WILL BE STORED IN LOC "FW-REV" FOR EXAMINATION
000245                          *    BY USERS WITH NO CONSOLE.
000246                          *
000247                          *
000248                          *
000249                          *    THE PROGRAM WILL RUN APPROXOMATELY 3 - 5 MINUTES IF THERE ARE NO
000250                          *    HARDWARE FAULTS. IT WILL THEN DISPLAY:
000251                          *
000252                          *         PASS
000253                          *
000254                          *    AND CONTINUE EXECUTING. FOR SYSTEMS WITH NO CONSOLE
000255                          *    THE PROGRAM WILL HALT AT THE END OF EACH PASS WITH THE P
000256                          *    COUNTER DISPLAY = HEX 100.
000257                          *
000258                          *
000259                          *    TO CHANGE TO A SECOND MLCP ADDRESS USE MODE "I" OR RESTART THE
000260                          *    PROGRAM AT HEX 105. PROGRAM OPERATION WILL THEN PROCEED AS
000261                          *    AFTER AN INITIAL START AT HEX 100.
000262                          *
000263                          *
000264                          *    HITTING 'BREAK' DURING THE TEST WILL CAUSE THE PROGRAM TO
000265                          *    RETURN TO "NEXT".
000266                          *
000267                          *  ERROR REPORTS
000268                          *
000269                          *    AN ERROR MESSAGE WILL BE DISPLAYED IF A CONSOLE IS PRESENT.
000270                          *    ALL ERROR MESSAGES ARE AN INDICATION THAT THE COMM-
```

```
000271                          *    UNICATIONS PROCESSOR IS FAULTY.
000272                          *
000273                          *    ERROR DISPLAYS ARE AS FOLLOWS:
000274                          *
000275                          *              ERR MBXX AT YYYY
000276                          *
000277                          *              B7  B6  B5  B4  B3  B2  B1  I
000278                          *              R7  R6  R5  R4  R3  R2  R1  M
000279                          *
000280                          *         XX   = TEST LABLE. SEE JUMP TABLE AT LOCATION "BKK1"
000281                          *                FOR A LIST OF TESTS PERFORMED.
000282                          *         YYYY = ERROR LOCATION IN LISTING. HAS COMMENT
000283                          *                GIVING FAILING FUNCTION.
000284                          *
000285                          *         B1-B7, I, R1-R7, M ARE  CONTENTS OF REGISTERS.
000286                          *         INTERPERTATION OF THESE IS FOR SPECIALIST USAGE.
000287                          *
000288                          *         IN ALL CASES:
000289                          *
000290                          *              R3 = CHANNEL NUMBER
000291                          *
000292                          *         IN GENERAL
000293                          *
000294                          *              R6 = SHOULD BE DATA
000295                          *              R5 = ACTUAL DATA
000296                          *              R7 = WORD NUMBER IN BLOCK TRANSFER
000297                          *
000298                          *
000299                          *    A PROGRAM HALT WITH P DISPLAYING  A VALUE LESS THAN
000300                          *    HEX 100 INDICATES THAT A TRAP OCCURED. P DISPLAYS ONE
000301                          *    BEYOND THE ACTUAL TRAP VECTOR WHICH WAS TAKEN. THE
000302                          *    TRAP SAVE AREA IS FROM LOC. 2 - 9. REFER TO LEVEL 6
000303                          *    T + V OPERATING INSTRUCTION MANUAL, DOC
000304                          *    AW 94, SECTION 1.1 .
000305                          *
000306                          *    IF DURING THE OPERATION OF THE PROGRAM THE DISPLAY PANEL
000307                          *    BECOMES "HUNG" SUCH THAT THE DISPLAYED REGISTER IS
000308                          *    UNCHANGING AND NO OTHER REGISTER MAY BE SELECTED, THEN
000309                          *    THE MLCP BEING TESTED SHOULD BE REPLACED.
000310                          *
000311                          *
000312                          *
000313                          *
000314                          ****************************************
000315                          ****************************************
000316                          *
000317                          * RESTRICTIONS;
000318                          *
000319                          *    FOR SYSTEMS WITH FIRMWARE REV NUMBER 4 OR LESS, REV. A
000320                          *    OF THE PROGRAM MUST BE USED.
000321                          *
000322                          *    THE FIRMWARE REV. NUMBER AT THE TIME OF RELEASING
000323                          *    REV. B OF THIS PROGRAM IS "5".
000324                          *
000325                          *    THE FIRMWARE REV NUMBER AT THE TIME OF RELEASING REV C
000326                          *    OF THE PROGRAM IS "8".
000327                          *
000328                          *
000329                          *    THE FIRMWARE REV. NUMBER AT THE TIME OF RELEASING REV D
000330                          *    OF THE PROGRAM IS "9". REV D CONTAINS NO FUNCTIONAL
000331                          *    DIFFERENCES FOR SAF MODE.
000332                          *
000333                          *    REV F OF THE PROGRAM HAS BEEN TESTED ON MLCP REV 9
000334                          *    AND REV 11 FIRMWARE.
000335                          *
000336                          *
000337                          *    IF THIS PROGRAM IS TO BE RUN ON A SYSTEM WITH A BASIC CONTROL
000338                          *    PANEL, A SYSTEM CONSOLE MUST BE PRESENT.
000339                          *
000340                          ************************************
000341                          ************************************
000342                          *
000343                          *
000344                          ************************************************************************
```

```
000345                                  /
000346                                  *
000347                                  *
000348                                  *       STANDARD REGISTER ASSIGNMENTS
000349                                  *
000350                                  *             $B5          V$LIB ENTRY
000351                                  *             $B2          ENTRY TO MAJOR TEST
000352                                  *             $B4,$B1      ENTRY TO SUB-ROUTINES
000353                                  *E    $B7E       ENTRY TO ERROR ROUTINE
000354                                  *
000355                                  *E    $R3          CHANNEL
000356                                  *E    $R5E       'IS' DATA WORD
000357                                  *E    $R6E       'SHOULD BE' DATA WORD
000358                                  *E    $R7E       TABLE COUNT
000359                                  *
000360                                  *
000361        0000                      ZERO  EQU   $
000362                                        XLOC  ZHNISA,ZHIH15,ZV$HK
000363                                        XLOC  ZV$RD,ZHISA,ZHIH17
000364                                        XLOC  ZHISAZ,ZHIAFB
000365                                        XLOC  ZHWDTC,ZHRICC
000366                                        XLOC  ZHRIC1,ZHRICL
000367                                        XLOC  ZHPFR,ZHCOMM
000368                                        XLOC  ZV$IA,ZV$ITY,ZV$BRK,ZV$BKF
000369                                        CTRL  LINK ZV$TH
000370                                  *
000371                                  *
000372                                  *
000373                                  *
000374                                  *
000375                                  *
000376  00FF                                  ORG   ZERO+X'FF'
000377  00FF  0F06                      STOP  NOP   >RESTRT                   NO ERROR FOUND IN MLCC
000378                                  *
000379                                  *             OPERATOR MUST ENTER MLCC CHANNEL
000380                                  *
000381  0100  89C0 1225                STRT  CMZ   FRST                      CHECK FIRST  TIME FLAG
000382  0102  0981 010A                      BNE   STLOOP                    BRANCH IF NOT FIRST TIME
000383  0104  0F7C                           NOP   >STRT
000384                                  *
000385                                  * RESTART POINT
000386  0105  8700 1328                RESTRT CL   <PASS
000387  0107  0F82                      DRSR1 B     >DRSR                   BECOMES NOP AFTER FIRST TIME
000388  0108  0F88                            B     >RST                    BYPASS ZV$RD
000389                                  *
000390                                  DRSR  CALL  ZV$RD,MESG
        0109  FBC0 0003
        010B  D380 0000       X
        010D  0F80
        010E  145B
000391  010F  9870 0F02                      LDR   $R1,=Z'0F02'            NOP
000392  0111  9F00 0107                      STR   $R1,<DRSR1
000393  0113  8980 1326                RST   CMZ   <FRST
000394  0115  0980 012A                      BNE   <RST1                     BRANCH IF NOT FIRST1 TIME
000395  0117  1C3C                           LDV   $R1,=60
000396  0118  9F00 13BB                      STR   $R1,<HRTZ               DEFAULT FREQ IS 60
000397  011A  8C51                           STS   =$R1
000398  011B  82D1                           LB    =$R1,=Z'2000'
        011C  2000
000399  011D  050D                          BBT   >RST1                   B = 6/4X,5X
000400                                  *
000401                                        CALL  ZV$GC,MESG5
        011E  FBC0 0003
        0120  D380 0000       X
        0122  0F80
        0123  1470
000402                                        CALL  ZV$ID,HRTZ              INPUT FREQUENCY
        0124  FBC0 0003
        0126  D380 0000       X
        0128  0F80
        0129  13BB
000403                                  RST1  CALL  ZV$GC,MESG1
        012A  FBC0 0003
        012C  D380 0000       X
        012E  0F80
        012F  13C7
000404                                        CALL  ZV$IH,DUMMY
        0130  FBC0 0003
        0132  D380 0000       X
        0134  0F80
        0135  1341
000405  0136  9800 1341                      LDR   $R1,<DUMMY
000406  0138  1D01                           CMV   $R1,=1
000407  0139  0987                           BNE   >JP1
000408                                        CALL  ZV$PCH
        013A  FBF0 0001
        013C  D380 0000       X
000409  013E  0F80 0105                      B     <RESTRT
000410  0140  9570 FFC0                JP1   AND   $R1,=Z'FFC0'              STRIP TO CHANNEL NUMBER ONLY
000411  0142  CB80 0000       X              LAB   $B4,<ZHISA                SET UP FOR TRAP ON
000412  0144  CF80 0000       X              STB   $B4,<ZHNISA               UNAVAILIABLE RESOURCE
000413  0146  CB80 0160                      LAB   $B4,<NODEVF
000414  0148  CF80 0000       X              STB   $B4,<ZHIH15
000415  014A  9970 03C0                      CMR   $R1,=X'3C0'               COMPARE CHANNEL WITH HEX 3C0
000416  014C  0302                           BG    >CON1                     BRANCH IF GREATER
000417  014D  0F8A                           B     >JP2                      CHANNEL NUMBER TO SMALL
000418  014E  1E26                    CON1   ADV   $R1,=X'26'                PUT FUNCTION IN
000419  014F  8055                           IO    =$R5,=$R1                 INPUT DEVICE NUMBER
        0150  0051
000420  0151  C855                           LDR   $R4,=$R5
000421  0152  D570 FF00                      AND   $R5,=Z'FF00'
000422  0154  D970 2100                      CMR   $R5,=Z'2100'              COMPARE DEVICE NUMBER RESPONSE
000423  0156  090C                           BE    >CON2
000424                                  *
000425                                  JP2   CALL  ZV$ER,$,ERMG1             MLCC NOT FOUND ON THIS ADDRESS
        0157  FBC0 0003
        0159  D380 0000       X
        015B  0F80
        015C  0157
        015D  13D7
000426  015E  0F80 0113                      B     <RST
000427                                  *
000428  0160  8755                    NODEVF CL   =$R5                      TRAP HANDLER ROUTINE
```

```
000429  0161  0003                        RTT
000430                              *
000431                              *
000432                              *
000433                              *       PREPARE ACTIVE LINE TABLE
000434                              *
000435                              *
000436                              CON2    CALL    ZV$F,AILT,ALLONE,C16
        0162  FBC0 0003
        0164  D380 0000       X
        0166  0F80
        0167  1329
        0168  1324
        0169  1325
000437  016A  8700 1343                    CL      <CH-CT                  CHANNEL COUNT
000438  016C  8700 1339                    CL      <IMASK                  SET LINE MASK WORD TO 0
000439                              *
000440  016E  D854                        LDR     $R5,=$R4
000441  016F  D851                        LDR     $R3,=$R1
000442  0170  D570 03C0                    AND     $R3,=X'3C0'             STRIP $R3 TO SUBCH. NUMBER
000443  0172  3046                        SOR     $R3,6
000444  0173  9570 FC3F                    AND     $R1,=Z'FC3F'            STRIP TO CONTROL WORD WITHOUT SUBCH.
000445  0175  DF30 1329             CON2A  STR     $R5,<AILT.$R3           STORE ID IF SUBCH. PRESENT
000446  0177  BF00 1347                    STR     $R3,<IPR
000447  0179  D970 2178                    CMR     $R5,=X'2178'
000448  017B  09B0                        BNE     >CON2C
000449                              *
000450                              * READ EXTENDED ID
000451                              *
000452  017C  C851                        LDR     $R4,=$R1
000453  017D  4E02                        ADV     $R4,=2                  FORM FC FOR EXT ID
000454  017E  C380 10EC                    LNJ     $B4,<CQSCH
000455  0180  8055                        IO      =$R5,=$R4
        0181  0054
000456  0182  0703                        BIOT    >$+2+$AF
000457  0183  F380 11CB                    LNJ     $B7,<ERROR              INPUT EXTEND ID WAS NAK'ED
000458  0185  DF30 1329                    STR     $R5,<AILT.$R3
000459  0187  3B80 01AB                    BUDD    $R3,<CON2C
000460                                    CALL    ZV$IC,IDMSG             LINE
        0189  FBC0 0003
        018B  D380 0000       X
        018D  0F80
        018E  13CC
000461  018F  3041                        SOR     $R3,1                   GET LINE NUMBER
000462  0190  BF00 2B00                    STR     $R3,<RTB
000463                                    CALL    ZV$ID,RTB               NUMBER
        0192  FBC0 0003
        0194  D380 0000       X
        0196  0F80
        0197  2B00
000464  0198  B800 1347                    LDR     $R3,<IPR                GET BACK CHANNEL
000465                                    CALL    ZV$I,EQ                 EQUALS
        019A  FBC0 0003
        019C  D380 0000       X
        019E  0F80
        019F  13D0
000466  01A0  5048                        SOR     $R5,8                   ALIGN EXTENDED ID
000467  01A1  D470 2100                    OR      $R5,=Z'2100'            PUT IN MLCP ADDRESS
000468  01A3  BF00 1347                    STR     $R5,<IPR
000469                                    CALL    ZV$IM,IPR               PRINT ID
        01A5  FBC0 0003
        01A7  D380 0000       X
        01A9  0F80
        01AA  1347
000470  01AB  8930 1339             CON2C  LBT     <IMASK,$R3              SET MASK WORD OF ACTIVE LINES
000471  01AD  8AD3                  CON2B  INC     =$R3
000472  01AE  3D10                        CMV     $R3,=16                 SEE IF ALL SUBCH. TESTED
000473  01AF  0909                        BE      >CON3
000474  01B0  C851                        LDR     $R4,=$R1
000475  01B1  C380 10EC                    LNJ     $B4,<CQSCH
000476  01B3  8055                        IO      =$R5,=$R4               INPUT ID
        01B4  0054
000477  01B5  0701 FFBF                    BIOT    CON2A
000478  01B7  0FF6                        B       >CON2B
000479                              *
000480                              *
000481  01B8  2CF5                  CON3   LDV     $R2,=-11                NUMBER OF I/O TO BE CHANGED
000482  01B9  DB80 13C7                    LAB     $B5,<CONT1+11           PUT ADDRESS OF CONTROL TABLE IN $B5
000483  01BB  C870 003F             ALL    LDR     $R4,=X'3F'              LOAD MASK TO CLEAR CHANNEL
000484  01BD  9570 FC00                    AND     $R1,=Z'FC00'            CLEAR SUBCH. & FUNCTION
000485  01BF  9F00 133D                    STR     $R1,<MLCADD             STORE MLC ADDRESS
000486  01C1  C525                        AND     $R4,$B5,$R2             CLEAR CHANNEL
000487  01C2  9454                        OR      $R1,=$R4                PUT CHANNEL NUMBER IN
000488  01C3  9F25                        STR     $R1,$B5,$R2             STORE IT BACK IN CONTROL TABLE
000489  01C4  27F7                        BINC    $R2,>ALL
000490                              *
000491                              * ASK FOR NEXT, A, T, I, OR Q
000492                              *
000493  01C5  8700 134F                    CL      <PRTFLG
000494  01C7  8700 134D                    CL      <QFLG
000495                              RUTBG   CALL    ZV$QC, MESG2            ASK NEXT
        01C9  FBC0 0003
        01CB  D380 0000       X
        01CD  0F80
        01CE  020A
000496                                    CALL    ZV$IA,MASK,IPR
        01CF  FBC0 0003
        01D1  D380 0000       X
        01D3  0F80
        01D4  135E
        01D5  1347
000497  01D6  9800 1347                    LDR     $R1,<IPR
000498  01D8  1048                        SOR     $R1,8
000499  01D9  9970 0041                    CMR     $R1,=X'41'              A
000500  01DB  0900 020D                    BE      <SILOOP
000501  01DD  9970 0054                    CMR     $R1,=X'54'              P
000502  01DF  0985                        BNE     >RUT1
000503  01E0  8A80 134F                    INC     <PRTFLG                 SET PRINT FLAG
000504  01E2  0F80 01C9                    B       <RUTBG
000505                              *
000506  01E4  9970 0049             RUT1   CMR     $R1,=X'49'              I
000507  01E6  0900 0105                    BE      <RESTRT
000508  01E8  9970 0051                    CMR     $R1,=X'51'              Q
```

```
000509  01EA  0985                        BNE     >RUT3
000510  01EB  8A80 134D                   INC     <QFLG
000511  01ED  0F80 01C9                   B       <RUTBG
000512  01EF  9970 0046           RUT3    CMR     $R1,=X'46'
000513  01F1  0986                        BNE     >RUT3A
000514  01F2  8753                        CL      =$R3
000515  01F3  C380 1400                   LNJ     $B4,<PREV               PRINT FIRMWARE OPT, REV
000516  01F5  0F80 01C9                   B       <RUTBG
000517                            *
000518  01F7  9970 0052           RUT3A   CMR     $R1,=X'52'              R
000519  01F9  0985                        BNE     >RUT3B
000520  01FA  8A80 134E                   INC     <RFLG
000521  01FC  0F80 0233                   B       <BKK9A
000522                            RUT3B   CALL    ZV$IC,NOGO
        01FE  FBC0 0003
        0200  D380 0000       X
        0202  0F80
        0203  0206
000523  0204  0F80 01C9                   B       <RUTBG
000524                            *
000525  0206  494E 5641 4C49      NOGO    TEXT    'INVALID$'
        0209  4424
000526  020A  4E45 5854 2400      MESG2   TEXT    'NEXT$'
000527                            *
000528                            *
000529                            *
000530                            *
000531                            *
000532  020D  8700 1327           STLOOP  CL      <LOOP                   SET LOOP COUNT TO 0
000533  020F  8700 13B6                   CL      <SEC                    CLEAR SECONDS COUNTER
000534  0211  8700 13B7                   CL      <TTOT                   CLEAR TIME TOTALIZER
000535  0213  0F82                        B       >BKK1
000536                            *
000537                            * JUMP TABLE FOLLOWS. THIS IS THE CONTROL SECTION OF THE PROGRAM.
000538                            *
000539                            *                                       LABLE  -  DESCRIPTION
000540  0214  0F03                NOP     NOP     >BKK3
000541  0215  A380 0263           BKK1    LNJ     $B2,<RGNT               A - GO TO RANGE TEST
000542  0217  A380 031E           BKK3    LNJ     $B2,<CCBRT              B - GO TO CCB READ TEST
000543  0219  A380 0399           BKK4    LNJ     $B2,<MEMT               C - GO TO MEMORY TEST
000544  021B  A380 0474           BKK4A   LNJ     $B2,<MNKT               D - GO TO MEMORY NAK TEST
000545  021D  A380 04A8           BKK4B   LNJ     $B2,<LCIOB              E - GO TO OUTPUT BYTE IN LCI TEST
000546  021F  A380 054C                   LNJ     $B2,<SITEST             S1 - SOFT INITIALIZE TEST
000547  0221  A380 05CD           BKK5    LNJ     $B2,<CHPGT              F - GO TO CHANNEL PROGRAM TEST
000548  0223  A380 064F           BKK6    LNJ     $B2,<INSTT              G - GO TO INSTRUCTION TEST
000549  0225  A380 06BD           BKK7    LNJ     $B2,<INCONT             H - GO TO OUTPUT INTERRUPT CONTROL TEST
000550  0227  A380 06EF           BKK7A   LNJ     $B2,<IZTST              J - GO TO INITIALIZE TEST
000551  0229  A380 0803           BKK7B   LNJ     $B2,<STIORC             K - STOP IO  TEST
000552  022B  A380 0856           BKK7C   LNJ     $B2,<RCO                L - RECEIVE TEST
000553  022D  A380 089E           BKK7D   LNJ     $B2,<TRAN1              M - TRANSMIT TESTS
000554  022F  A380 0AB2           BKK8    LNJ     $B2,<CRCT               N - GO TO DATA INPUT AND CRC TEST
000555  0231  A380 0B94           BKK9    LNJ     $B2,<OTPT               P - GO TO OUTPUT OUTPUT TEST
000556  0233  A380 0C0B           BKK9A   LNJ     $B2,<INST1              R - FIRST INTERRUPT TEST
000557  0235  A380 0CB1           BKK9B   LNJ     $B2,<DSST               S - DATA SET SCAN TEST
000558  0237  A380 0DF6           BKK10   LNJ     $B2,<SITST              T - DEFERRED INTERRUPT TEST
000559                            * ACCUMULATE TIME
000560  0239  9800 13B6                   LDR     $R1,<SEC                GET NUMBER OF SECONDS
000561  023B  9970 0078                   CMR     $R1,=120                CHECK IF 3 MIN
000562  023D  028A                        BGE     >PSPT                   BRANCH MEANS > OR = 3 MIN
000563  023E  8980 134D                   CMZ     <QFLG
000564  0240  0987                        BNE     >PSPT                    B = "Q" MODE.
000565  0241  8980 134E                   CMZ     <RFLG
000566  0243  0980 0233                   BNE     <BKK9A                  B = DO ONLY RUPT TESTS
000567  0245  0F80 0217                   B       <BKK3
000568                            PSPT    CALL    ZV$IC,PASMSG
        0247  FBC0 0003
        0249  D380 0000       X
        024B  0F80
        024C  025F
000569  024D  8A80 1328                   INC     <PASS                   INCREMENT PASS COUNT
000570  024F  8980 134D                   CMZ     <QFLG                   CHECK QUICK FLAG
000571  0251  0980 01C9                   BNE     <RUTBG                  BACK TO NEXT IF SET
000572  0253  8980 134E                   CMZ     <RFLG                   CHECK RUPT MODE
000573  0255  0980 0233                   BNE     <BKK9A                  B = SET
000574                            *
000575  0257  8980 0000       X           CMZ     <ZV$TTY
000576  0259  0980 0100                   BNE     <STRT                   B = CONSOLE ON SYSTEM
000577                            *
000578  025B  8700 00FF           NOTTY   CL      <STOP                   FIRST PASS
000579  025D  0F80 00FF                   B       <STOP
000580  025F  2050 4153 5320      PASMSG  TEXT    ' PASS $'
        0262  2400
000581                            *
000582                            *-----------------------------------------------------------------
000583                            *
000584                            *           RANGE TEST
000585                            *
000586                            *
000587  0263  9870 4120           RGNT    LDR     $R1,=A'A '
000588  0265  C380 13E4                   LNJ     $B4,<PLB
000589  0267  9380 0EC9                   LNJ     $B1,<INTIIZ             INITIALIZE TRAP + RUPT ERROR VECTORS
000590  0269  C380 106D                   LNJ     $B4,<GENIIZ             GENERAL INITIALIZE TO MLCC
000591                            *
000592  026B  9870 0300                   LDR     $R1,=X'300'             LOAD IMGA WITH HEX 300
000593  026D  9F00 1357                   STR     $R1,<IMGA+($AF-1)
000594  026F  8751                        CL      =$R1                    CLEAR $R1
000595  0270  8753                        CL      =$R3                    FIRST CHANNEL
000596  0271  C380 105D           NXCH    LNJ     $B4,<FLN                FIND LINE ON
000597  0273  0FA4                        B       >JMPEXT                 GO DO SECOND HALF OF TEST
000598                            *
000599  0274  6C01                        LDV     $R6,=1                  INITIALIZE RANGE
000600  0275  4C0A                        LDV     $R4,=10
000601  0276  CA00 13B6                   ADD     $R4,<SEC                UPDATE TIME, 10 SEC/CHAN
000602  0278  CF00 13B6                   STR     $R4,<SEC
000603  027A  C800 13BC           NXRG    LDR     $R4,<CONT1              LOAD $R4 WITH I/O CONTROL WORD
000604  027C  C380 10EC                   LNJ     $B4,<CGSCH              PUT CHANNEL IN I/O CONTROL WORD
000605  027E  81C8 10D8                   IOLD    *IMGA,=$R4,=$R6         OUTPUT ADDRESS AND RANGE
        0280  0054
        0281  0056
000606  0282  0703                        BIOT    >IOI1
000607  0283  F380 11CB                   LNJ     $B7,<ERROR              I/O TRANSFER DID NOT TAKE PLACE
000608                            *
000609  0285  4E06                IOT1    ADV     $R4,=6                  OUTPUT CCB CONTROL FIELD
```

```
000610  0286 8070 0000                    IO      =0,=$R4
        0288 0054
000611  0289 0703                        BIOT    >$+2+$AF
000612  028A F380 11CB                    LNJ     $B7,<ERROR
000613                              *
000614  028C C800 13BF                    LDR     $R4,<CONT4          INPUT NXT STAT TO ADVANCE POINTER
000615  028E C360 10EC                    LNJ     $B4,<CGSCH
000616  0290 8040 10B0                    IO      DUMMY,=$R4          INPUT NXT STAT
        0292 0054
000617  0293 0F85                        B       >IOI3
000618  0294 F380 11CB                    LNJ     $B7,<ERROR          INSTRUCTION WAS NACKED
000619  0296 0F82                        B       >IOI3
000620  0297 0FA1                  JMPEXT B      >SCCBTS
000621                              *
000622  0298 C800 13BD              IOI3  LDR     $R4,<CONT2          LOAD $R4 WITH I/O CONTROL WORD
000623  029A C360 10EC                    LNJ     $B4,<CGSCH          I/O CONTROL WORD IN $R4
000624  029C 8055                         IO      =$R5,=$R4           INPUT RANGE
        029D 0054
000625  029E 0703                        BIOT    >IOI2
000626  029F F380 11CB                    LNJ     $B7,<ERROR          I/O TRANSFER DID NOT TAKE PLACE
000627  02A1 E955                   IOI2  CMR     $R6,=$R5            COMPARE OUTPUT RANGE WITH INPUT RANGE
000628  02A2 0993                        BNE     >RGE                NOT EQUAL BRANCH TO RGE
000629                              *
000630                              *
000631  02A3 8980 134D              RGNT1 CMZ     <QFLG
000632  02A5 0903                        BE      >RGNT2              B = NOT QUICK MODE
000633  02A6 6001                        SOL     $R6,1
000634  02A7 0F82                        B       >RGNT3
000635                              *
000636  02A8 8AD6                   RGNT2 INC     =$R6                SET FOR NEXT RANGE
000637  02A9 06D1                   RGNT3 BCF     >NXRG               IF CARRY FALSE THEN BRANCH TO NXRG
000638  02AA 8AD3                         INC     =$R3                ADD ONE TO CHANNEL
000639                              *
000640                              * TEST FOR BREAK
000641                              *
000642                                    CALL    ZV$BRK
        02AB F0F0 0001
        02AD D380 0000              X
000643  02AF 8980 0000              X           CMZ     <ZV$BKF
000644  02B1 0980 01C9                    BNE     <RUTBG
000645  02B3 0F80 0271                    B       <NXCH               BRANCH TO NXCH
000646                              *
000647  02B5 F380 11CB              RGE   LNJ     $B7,<ERROR          RANGE ERROR
000648  02B7 0FEC                         B       >RGNT1              CONTINUE TESTING
000649                              *
000650                              * TEST CCB'S DON'T OVERWRITE EACH OTHER
000651                              *
000652  02B8 C380 106D              SCCBTS LNJ    $B4,<GENIT2         GENERAL INITIALIZE
000653  02BA 2C01                        LDV     $R2,=1              INITIAL DATA
000654  02BB F870 0200                   LDR     $R7,=X'200'         DATA FOR CONTROL FLAG
000655  02BD 8753                        CL      =$R3
000656  02BE C380 105D              NLINE LNJ     $B4,<FLN
000657  02C0 0FA0                        B       >RCCBS              NO MORE LINES
000658  02C1 1CFC                        LDV     $R1,=-4             COUNTS 4 CCB'S
000659  02C2 C800 13BC              NXTCCB LDR    $R4,<CONT1          IOLD FUNCTION CODE
000660  02C4 C380 10EC                    LNJ     $B4,<CGSCH          FORM I/O CONTROL WORD
000661  02C6 81C8 1090                    IOLD    *IMGA,=$R4,=$R2
        02C8 0054
        02C9 0052
000662  02CA 0703                        BIOT    >$+2+$AF
000663  02CB F380 11CB                    LNJ     $B7,<ERROR          IOLD WAS NACKED
000664  02CD 4E06                         ADV     $R4,=6              FORM CCB CONTROL WORD
000665  02CE 8057                         IO      =$R7,=$R4           OUTPUT CONTROL FLAG
        02CF 0054
000666  02D0 0703                        BIOT    >$+2+$AF
000667  02D1 F380 11CB                    LNJ     $B7,<ERROR
000668  02D3 8AD2                         INC     =$R2                BUMP FOR NEXT RANGE
000669  02D4 FA70 0100                    ADD     $R7,=X'100'         FORM NEXT CONTROL WORD
000670  02D6 17EC                         BINC    $R1,>NXTCCB         BRANCH MEANS THIS CHAN NOT DONE
000671  02D7 81C8 107F                    IOLD    *IMGA,=$R4,=$R2     GIVE  A 5TH IOLD
        02D9 0054
        02DA 0052
000672  02DB 0783                        BIOF    >$+2+$AF            IT SHOULD BE NAK'ED
000673  02DC F380 11CB                    LNJ     $B7,<ERROR          5TH IOLD WASN'T NAK'ED
000674  02DE 8AD3                         INC     =$R3                BUMP CHANNEL LINE
000675  02DF 0FDF                         B       >NLINE
000676                              *
000677                              * READ RANGES SET UP
000678                              *
000679  02E0 2C01                   RCCBS LDV     $R2,=1              INITIAL DATA
000680  02E1 F870 0202                   LDR     $R7,=X'202'         DATA FOR STATUS
000681  02E3 8753                        CL      =$R3
000682  02E4 C380 105D              NL1   LNJ     $B4,<FLN            FIND LINE NUMBER
000683  02E6 6382                        JMP     $B2                 ALL DONE, EXIT TEST
000684  02E7 1CFC                        LDV     $R1,=-4             COUNTER FOR 4 CCB'S
000685  02E8 C800 13BF              RNXR  LDR     $R4,<CONT4
000686  02EA C380 10EC                    LNJ     $B4,<CGSCH          FORM IO CONTROL WORD
000687  02EC 8040 1054                    IO      DUMMY,=$R4          INPUT NEXT STATUS
        02EE 0054
000688  02EF 0703                        BIOT    >$+2+$AF
000689  02F0 F380 11CB                    LNJ     $B7,<ERROR          INPUT NXT STATUS WAS NAK'ED
000690  02F2 C800 13BD                    LDR     $R4,<CONT2
000691  02F4 C380 10EC                    LNJ     $B4,<CGSCH          FORM IO CONTROL WORD
000692  02F6 8055                         IO      =$R5,=$R4           INPUT RANGE
        02F7 0054
000693  02F8 0703                        BIOT    >$+2+$AF
000694  02F9 F380 11CB                    LNJ     $B7,<ERROR
000695  02FB E852                         LDR     $R6,=$R2            GET SHOULD-BE DATA TO R6
000696  02FC D956                         CMR     $R5,=$R6
000697  02FD 0903                        BE      >$+2+$AF
000698  02FE F380 11CB                    LNJ     $B7,<ERROR          RANGE READ IN IS WRONG
000699                              * DO AN INPUT STATUS. IF IT ADVANCES THE INPUT POINTER A FAILURE WILL RESULT.
000700  0300 C800 13C1                    LDR     $R4,<CONT6          INPUT STATUS CFUNCTION
000701  0302 C380 10EC                    LNJ     $B4,<CGSCH          FORM I/O CONTROL WORD
000702  0304 8055                         IO      =$R5,=$R4           INPUT STATUS
        0305 0054
000703  0306 0703                        BIOT    >$+2+$AF
000704  0307 F380 11CB                    LNJ     $B7,<ERROR          INPUT STATUS WAS NAK'ED
000705  0309 E857                         LDR     $R6,=$R7            GET SHOULD BE DATA TO R6
000706  030A D956                         CMR     $R5,=$R6
000707  030B 0903                        BE      >$+2+$AF
000708  030C F380 11CB                    LNJ     $B7,<ERROR          STATUS READ IN INCORRECTLY
000709                              *
```

```
000710  030E  8AD2                        INC   =$R2                 BUMP RANGE
000711  030F  FA70 0101                   ADD   $R7,=X'0101'         FORM NEXT STATUS
000712  0311  17D7                        BINC  $R1,>RNXR            READ NEXT RANGE
000713                              *
000714                              *EGIVE A 5TH INPUT NEXT STATUS AND CHECK ITS NAK'ED
000715                              *
000716  0312  C800 13BF                   LDR   $R4,<CONT4           FORM I/O CONTROL
000717  0314  C380 10EC                   LNJ   $B4,<CGSCH           WORD FOR INPUT NEXT STATUS
000718  0316  8055                        IO    =$R5,=$R4            SHOULD BE NAK'ED
        0317  0054
000719  0318  0783                        BIOF  >$+2+$AF
000720                              *
000721  0319  F380 11CB                   LNJ   $B7,<ERROR           INPUT NEXT STATUS NOT NAK'ED
000722                              *
000723  031B  8AD3                        INC   =$R3
000724  031C  0F81 FFC7                   B     NL1                  FIND NEXT LINE NUMBER
000725                              *-------------------------------------------------------------
000726                              *
000727                              *
000728                              *
000729                              *           CCB READ TEST - TESTS OUTPUT OF CCB & BLOCK READ
000730                              *
000731                              *
000732  031E  9870 4220             CCBRT LDR   $R1,=A'B '
000733  0320  C380 13E4                   LNJ   $B4,<PLB
000734  0322  8753                        CL    =$R3
000735  0323  C380 106D             NXCB  LNJ   $B4,<GENIIZ          GENERAL INITIALIZE
000736                              *
000737  0325  9800 1359                   LDR   $R1,<SBCK+$AF-1      GET ADDRESS CONSTANT LSB
000738  0327  1001                        SOL   $R1,1                *2 TO GET BYTE ADDRESS
000739  0328  1058                        SCR   $R1,8                SWAP BYTES
000740  0329  9F00 135A                   STR   $R1,<SBCCB
000741  032B  8751                  NXCB1 CL    =$R1                 CLEAR INDEX FOR OUTPUT ADDRESS
000742  032C  8700 135E                   CL    <MASK                LOAD ZERO INTO MASK
000743  032E  4C08                        LDV   $R4,=8               LOAD $R4 WITH BYTE RANGE OF CCB
000744  032F  CF00 1352                   STR   $R4,<RANGE
000745  0331  5C04                        LDV   $R5,=4               LOAD RANGEW WITH WORD RANGE OF CCB
000746  0332  DF00 1354                   STR   $R5,<RANGEW
000747  0334  C380 105D                   LNJ   $B4,<FLN             FIND LINE ON
000748  0336  8382                        JMP   $B2                  NO LINE ON JUMP OUT OF TEST
000749  0337  BF00 2800                   STR   $R3,<RTB             RE-INITIALIZE RTB
000750  0339  BF00 2801                   STR   $R3,<RTB+1
000751  033B  BF00 2802                   STR   $R3,<RTB+2
000752  033D  BF00 2803                   STR   $R3,<RTB+3
000753                              *
000754  033F  D853                        LDR   $R5,=$R3             FIND ADDRESS FOR BLOCK READ
000755  0340  5F20                        MLV   $R5,=X'20'           R5 = CHAN X 20
000756  0341  DA70 0E10                   ADD   $R5,=X'E10'          LOCATION OF CCB TO READ
000757  0343  BF00 1347                   STR   $R3,<TPR             STORE CHAN NUMB
000758  0345  3B03                        BEVN  $R3,>CBRT3           BRANCH IF READ
000759  0346  88D3                        DEC   =$R3                 FORM READ CHANNEL
000760  0347  5EF8                        ADV   $R5,=-8              FORM CCB ADDRESS FOR WRITE CHANNEL
000761  0348  BF00 0398             CBRT3 STR   $R3,<READ            STORE READ CHANNEL
000762  034A  DF00 0351                   STR   $R5,<CBRT1
000763  034C  C800 1352                   LDR   $R4,<RANGE
000764                              *
000765  034E  C380 1102                   LNJ   $B4,<MCCB            FORM CCB FOR BLOCK READ
000766  0350  2800                        DC    <RTB
000767  0351  0000                  CBRT1 DC    X'0'                 RAM ADDRESS TO READ
000768                              *
000769                              *
000770  0352  B800 1347                   LDR   $R3,<TPR             GET CHANNEL UNDER TEST
000771  0354  C800 1352                   LDR   $R4,<RANGE           FORM CCB TO READ BACK
000772  0356  C380 1102             NXCB2 LNJ   $B4,<MCCB
000773  0358  13B4                        DC    <RM1NST
000774  0359  55AA                        DC    X'55AA'
000775                              *
000776  035A  B800 0398             NXCB3 LDR   $R3,<READ            GET READ CHANNEL NUMBER
000777  035C  C800 13C2                   LDR   $R4,<CONT7           LOAD $R4 WITH I/O CONTROL WORD
000778  035E  C380 10EC                   LNJ   $B4,<CGSCH           PUT I/O CONTROL WORD IN $R4
000779  0360  8070 0800                   IO    =X'800',=$R4         READ BLOCK IN FROM MLCC
        0362  0054
000780  0363  0703                        BIOT  >$+2+$AF
000781  0364  F380 11CB                   LNJ   $B7,<ERROR
000782                              *
000783                              * READ BACK STATUS AFTER BLOCK READ TO MAKE SURE IT'S CORRECT.
000784                              *
000785  0366  C380 117D                   LNJ   $B4,<DLAY            TIME DELAY
000786                              *
000787  0368  C380 10DA                   LNJ   $B4,<INXT            INPUT NEXT STATUS
000788  036A  E870 1000                   LDR   $R6,=X'1000'         SHOULD BE STATUS. CCB COMPLETE ON.
000789  036C  D956                        CMR   $R5,=$R6
000790  036D  0903                        BE    >$+2+$AF
000791  036E  F380 11CB                   LNJ   $B7,<ERROR           STATUS WRONG AFTER BLOCK READ
000792  0370  B800 1347                   LDR   $R3,<TPR             GET BACK CHANNEL UNDER TEST
000793                              *
000794                                    CALL  ZV$C,SBCCB,RT5,MASK,RANGEW,ERAR
        0372  FBC0 0003
        0374  D380 0000             X
        0376  0F80
        0377  135A
        0378  2800
        0379  135E
        037A  1354
        037B  1360
000795  037C  89C0 0FE3                   CMZ   ERAR                 IF ERROR HALT
000796  037E  090F                        BE    >NOER1
000797  037F  C800 0351                   LDR   $R4,<CBRT1           STARTING LOCATION OF CCB IN RAM
000798  0381  D800 1362                   LDR   $R5,<ERAR+2
000799  0383  E800 1361                   LDR   $R6,<ERAR+1
000800  0385  F800 1360                   LDR   $R7,<ERAR
000801  0387  F380 11CB                   LNJ   $B7,<ERROR           BLOCK READ ERROR
000802                                    CALL  ZV$C0                CONTINUE LOOKING FOR BLOCK READ ERROR
        0389  FBF0 0001
        038B  D380 0000             X
000803  038D  8980 1326             NOER1 CMZ   <FRST                CHECK IF FIRST TIME THRU
000804  038F  0981 0003                   BNE   BLUG
000805  0391  C380 1400                   LNJ   $B4,<PREV            PRINT FIRMWARE REVISION
000806  0393  8A80 1326             BLUG  INC   <FRST                SET FIRST TIME FLAG
000807  0395  8AD3                        INC   =$R3                 INCREMENT CHANNEL NUMBER
000808                              *
000809                              *
000810  0396  0F80 0323                   B     <NXCB
```

```
000811  0398  0000                READ    DC      0                       READ CHANNEL STORED HERE
000812                            *-------------------------------------------------------------------
000813                            *
000814                            *
000815                            *
000816                            *
000817                            *           MLCC MEMORY TEST
000818                            *
000819                            *
000820  0399  9870 4320           MEMT    LDR     $R1,=A'C '
000821  039B  C380 13E4                   LNJ     $B4,<PLB
000822  039D  8700 135E                   CL      <MASK                   INITIALIZE MASK FOR COMPARISON
000823  039F  C380 106D                   LNJ     $B4,<GENIT2             GENERAL INITIALIZE
000824                            *
000825  03A1  C870 5555                   LDR     $R4,=X'5555'            INITIALIZE MASK1 FOR RETURN BLOCK
000826  03A3  CF00 135F                   STR     $R4,<MASK1
000827  03A5  8753                       CL      =$R3
000828  03A6  C380 105D           MEMTA   LNJ     $B4,<FLN                FIND LINE ON
000829  03A8  8382                       JMP     $B2
000830                            *
000831  03A9  3B00 03AD                   BEVN    $R3,<MEMTB              TEST ON A LINE, NOT CHANNEL BASIS
000832  03AB  8AD3                       INC     =$R3
000833  03AC  0FFA                       B       >MEMTA                  FIND NEXT LINE
000834  03AD  8751           MEMTB   CL      =$R1                    CLEAR INDEX FOR OUTPUT ADDRESS
000835  03AE  8754                       CL      =$R4
000836  03AF  E380 03BE                   LNJ     $B6,<TESM               TEST MEMORY WITH ALL BITS NOT SET
000837  03B1  4CFF                       LDV     $R4,=-1
000838  03B2  E380 03BE                   LNJ     $B6,<TESM               TEST MEMORY WITH ALL BITS SET
000839  03B4  C870 0110                   LDR     $R4,=X'110'             TEST MEMORY BY FLOATING A
000840  03B6  E380 03BE                   LNJ     $B6,<TESM               ONE THROUGH A FIELD OF ZEROS
000841  03B8  C870 FEEF                   LDR     $R4,=Z'FEEF'            TEST MEMORY BY FLOATING A
000842  03BA  E380 03BE                   LNJ     $B6,<TESM               ZERO THROUGH A FIELD OF ONE'S
000843  03BC  8AD3                       INC     =$R3                    SET FOR NEXT CHANNEL
000844  03BD  0FE9                       B       >MEMTA                  TRY NEXT LINE
000845                            *
000846                            *
000847  03BE  2C01           TESM    LDV     $R2,=1                  INITIALIZE $R2
000848  03BF  E870 0300                   LDR     $R6,=X'300'             LOAD NUMBER OF WORDS INTO RANGEW
000849  03C1  EF00 1354                   STR     $R6,<RANGEW
000850  03C3  8700 1355                   CL      <RAMAD                  INITIALIZE RAM ADDRESS
000851                            *
000852  03C5  9380 044F                   LNJ     $B1,<SET                FILL READ AREA WITH SET PATTERN
000853                            *
000854  03C7  BB80 2400                   LAB     $B3,<SDB                LOAD $B3 WITH ADDRESS OF SDB
000855  03C9  E800 1354                   LDR     $R6,<RANGEW             INITIALIZE LOOP COUNTER
000856  03CB  8256                       NEG     =$R6
000857  03CC  CF73           FILL    STR     $R4,+$B3                STORE PATTERN INTO SDB
000858  03CD  4011                       SCL     $R4,1                   ROTATE TO FORM FULL PATTERN
000859  03CE  67FE                       BINC    $R6,>FILL
000860                            *
000861  03CF  BF00 1349                   STR     $R3,<TEMP1
000862  03D1  BF00 1347                   STR     $R3,<IPR
000863  03D3  3B85                       BODD    $R3,>MTM6               FIND WRITE SUBCH.
000864  03D4  8A80 1347                   INC     <IPR
000865  03D6  BE00 1347                   SWR     $R3,<IPR
000866  03D8  9380 0FDA           MTM6    LNJ     $B1,<SDATA              BLOCK WRITE DATA PATTERN TO RAM
000867  03DA  2400                       DC      <SDB
000868  03DB  0600                       DC      X'600'
000869  03DC  0200                       DC      X'200'                  RAM ADDRESS
000870  03DD  8000                       DC      Z'8000'                 EVEN BYTE ADDRESS, 250 MS DELAY
000871                            *
000872                            *
000873  03DE  BE00 1347                   SWR     $R3,<IPR                GET READ SUBCH.
000874  03E0  9380 1009                   LNJ     $B1,<RDATA              READ DATA PATTERN FROM RAM
000875  03E2  2800                       DC      <RIB
000876  03E3  0600                       DC      X'600'
000877  03E4  0200                       DC      X'200'                  RAM ADDRESS
000878  03E5  8000                       DC      Z'8000'                 EVEN BYTE, 250 MS DELAY
000879                            *
000880                            *
000881  03E6  9380 0458                   LNJ     $B1,<CRWB               COMPARE READ AND WRITE BLOCKS
000882  03E8  C870 0300                   LDR     $R4,=768                NUMBER OF BYTES TO CCB AREA
000883  03EA  CF00 1354                   STR     $R4,<RANGEW
000884                            *
000885  03EC  BE00 1347                   SWR     $R3,<IPR                GET WRITE SUBCH.
000886  03EE  9380 0FDA                   LNJ     $B1,<SDATA              WRITE DATA PATTERN
000887  03F0  2400                       DC      <SDB
000888  03F1  0600                       DC      1536
000889  03F2  0800                       DC      2048                    RAM ADDRESS
000890  03F3  8000                       DC      Z'8000'                 EVEN BYYTE,250 MS DELAY
000891                            *
000892                            *
000893  03F4  9380 044F                   LNJ     $B1,<SET                FILL READ AREA WITH SET PATTERN
000894                            *
000895  03F6  BE00 1347                   SWR     $R3,<IPR                GET READ SUBCH.
000896  03F8  9380 1009                   LNJ     $B1,<RDATA              READ DATA PATTERN FROM RAM
000897  03FA  2800                       DC      <RIB
000898  03FB  0600                       DC      1536
000899  03FC  0800                       DC      2048                    RAM ADDRESS
000900  03FD  8000                       DC      Z'8000'                 EVEN BYTE, 250 MS DELAY
000901                            *
000902                            *
000903  03FE  9380 0458                   LNJ     $B1,<CRWB               COMPARE READ AND WRITE BLOCKS
000904                            *
000905                            *
000906                            *
000907                            *
000908                            *   FIND RAM LOCATION AND NUMBER OF BYTES FOR
000909                            *      BLOCK READ AND WRITE
000910                            *
000911  0400  C853                       LDR     $R4,=$R3                COPY SUBCH.
000912  0401  4F20                       MLV     $R4,=X'20'
000913  0402  CF00 043D                   STR     $R4,<Q5                 DISTANCE FROM EOO TO CHANNEL CCP AREA
000914  0404  CF00 0447                   STR     $R4,<Q6                 FOR BLOCK READ AND WRITE
000915  0406  CA70 0E40                   ADD     $R4,=X'E40'
000916  0408  CF00 0420                   STR     $R4,<Q2                 ADDRESS ABOVE CCBS IN USE
000917  040A  CF00 042A                   STR     $R4,<Q4                 FOR BLOCK READ AND WRITE
000918  040C  C270 1000                   SUB     $R4,=X'1000'
000919  040E  8254                       NEG     =$R4
000920  040F  CF00 041F                   STR     $R4,<Q1                 STORE NUMBER OF BYTES FOR BLOCK
000921  0411  CF00 0429                   STR     $R4,<Q3                 READ AND WRITE BELOW CURRENT CHANNEL
000922  0413  4041                       SUR     $R4,1                   FIND AND STORE NUMBER OF WORDS
000923  0414  CF00 1354                   STR     $R4,<RANGEW             FOR BLOCK COMPARE
```

```
000924                          *
000925  0416  BE00 1347         *       SWR     $R3,<IPR                GET WRITE SUBCH.
000926  0418  9800 041F         *       LDR     $R1,<Q1                 SET RANGE
000927  041A  1A80 0433         *       BLEZ    $R1,<MTM10              SKIP NEXT TEST FOR LAST LINE
000928                          *
000929  041C  9380 0FDA         *       LNJ     $B1,<SDATA              WRITE DATA TO RAM
000930  041E  2400              *       DC      <SDB
000931  041F  0000              *Q1     DC      0                       RANGE = TOP - CCB TOP
000932  0420  0000              *Q2     DC      0                       RAM ADDRESS, CCB TOP
000933  0421  8000              *       DC      Z'8000'                 EVEN BYTE, 250 MS DELAY
000934                          *
000935                          *
000936                          *
000937  0422  9380 044F         *       LNJ     $B1,<SET                FILL READ BACK AREA WITH SET PATTERN
000938                          *
000939  0424  BE00 1347         *       SWR     $R3,<IPR                GET READ SUBCH.
000940  0426  9380 1009         *       LNJ     $B1,<RDATA              READ DATA PATTERN FROM RAM
000941  0428  2800              *       DC      <RTB
000942  0429  0000              *Q3     DC      0                       RANGE = TOP - CCB TOP
000943  042A  0000              *Q4     DC      0                       RAM ADDRESS, CCB TOP
000944  042B  8000              *       DC      Z'8000'                 EVEN BYTE, 250 MS DELAY
000945                          *
000946                          *
000947  042C  9380 0458         *       LNJ     $B1,<CRWB               COMPARE READ AND WRITE BLOCKS
000948                          *
000949  042E  D870 0E40         *       LDR     $R5,=X'E40'             SEE IF YOU HAVE TO TEST
000950  0430  D900 0420         *       CMR     $R5,<Q2                 BELOW CURRENT CCB
000951  0432  091A              *       BE      >TESM1
000952                          *
000953                          * TEST MEMORY FROM E00 TO CCB IN USE
000954                          *
000955  0433  C800 043D         *MTM10  LDR     $R4,<Q5                 FIND RANGE IN WORDS FOR
000956  0435  4041              *       SDR     $R4,1                   BLOCK COMPARE
000957  0436  CF00 1354         *       STR     $R4,<RANGEW
000958                          *
000959  0438  BE00 1347         *       SWR     $R3,<IPR                GET WRITE SUBCH.
000960  043A  9380 0FDA         *       LNJ     $B1,<SDATA              SEND DATA PATTERN TO RAM
000961  043C  2400              *       DC      <SDB
000962  043D  0000              *Q5     DC      0
000963  043E  0E00              *       DC      X'E00'
000964  043F  8000              *       DC      Z'8000'                 EVEN BYTE, 250 MS DELAY
000965                          *
000966                          *
000967  0440  9380 044F         *       LNJ     $B1,<SET                FILL READ BACK AREA WITH SET PATTERN
000968                          *
000969  0442  BE00 1347         *       SWR     $R3,<IPR                GET READ SUBCH.
000970  0444  9380 1009         *       LNJ     $B1,<RDATA              READ DATA PATTERN FROM RAM
000971  0446  2800              *       DC      <RTB
000972  0447  0000              *Q6     DC      0
000973  0448  0E00              *       DC      X'E00'                  RAM ADDRESS
000974  0449  8000              *       DC      Z'8000'                 EVEN BYTE, 250 MS DELAY
000975                          *
000976                          *
000977  044A  9380 0458         *       LNJ     $B1,<CRWB               COMPARE READ AND WRITE BLOCKS
000978                          *
000979  044C  B800 1349         *TESM1  LDR     $R3,<TEMP1              GET TESTED CHANNEL
000980  044E  8386              *       JMP     $B6                     RETURN FROM SUBROUTINE
000981                          *
000982                          *
000983                          *SET    CALL    ZV$F,RTB,DFLT,RANGEW
        044F  FBC0 0003
        0451  D380 0000         X
        0453  0F80
        0454  2800
        0455  133A
        0456  1354
000984  0457  8381              *       JMP     $B1
000985                          *
000986                          *
000987                          *      COMPARE SEND BLOCK WITH RETURN BLOCK
000988                          *
000989                          *CRWB   CALL    ZV$C,SDB,RTB,MASK,RANGEW,ERAR
        0458  FBC0 0003
        045A  D380 0000         X
        045C  0F80
        045D  2400
        045E  2800
        045F  135E
        0460  1354
        0461  1360
000990  0462  69C0 0EFD          *       CMZ     ERAR                    SEE IF ERROR PRESENT
000991  0464  090F              *       BE      >MBM                    IF PRESENT HALT WITH
000992  0465  C800 1355         *       LDR     $R4,<RAMAD              STARTING LOCATION OF BLOCK IN RAM
000993  0467  D800 1362         *       LDR     $R5,<ERAR+2             '15' IN $R5
000994  0469  E800 1361         *       LDR     $R6,<ERAR+1             'SHOULD BE' IN $R6
000995  046B  F800 1360         *       LDR     $R7,<ERAR               LOCATION NUMBER OF BLOCK IN $R7
000996  046D  F380 11CB         *       LNJ     $B7,<ERROR              RAM MEMORY ERROR
000997                          *       CALL    ZV$CO                   CONTINUE TESTING FOR ERRORS
        046F  FBF0 0001
        0471  D380 0000         X
000998  0473  8381              *MBM    JMP     $B1
000999                          *----------------------------------------------------------------
001000                          *
001001                          *
001002                          *
001003                          *
001004                          *         UNAVAILABLE RESOURCE TEST
001005                          *
001006                          *
001007  0474  9870 4420         *MNKT   LDR     $R1,=A'D '
001008  0476  C380 13E4         *       LNJ     $B4,<PLB                REPORT TEST
001009  0478  8753              *       CL      =$R3                    INITIALIZE CHANNEL
001010  0479  C380 106D         *       LNJ     $B4,<GENIIZ             INITIALIZE
001011                          *
001012  047B  C380 105D         *       LNJ     $B4,<FLN                FIND LINE ON
001013  047D  8382              *       JMP     $B2                     NO LINE ON JUMP OUT OF TEST
001014  047E  8751              *       CL      =$R1                    CLEAR INDEX FOR OUTPUT ADDRESS
001015  047F  5C20              *       LDV     $R5,=32                 LOAD RANGE WITH NUMBER OF BYTES
001016  0480  DF00 1352         *       STR     $R5,<RANGE
001017                          *       CALL    ZV$RD,ZV$HM             PUT MEMORY HIGH INTO $B7
        0482  FBF0 0001
        0484  D380 0000         X
001018  0486  C800 13BC         *       LDR     $R4,<CONT1              FORM I/O CONTROL WORD
```

```
001019  0488  C380 10EC              LNJ   $B4,<CGSCH
001020  048A  8187                   IOLD  $B7,=$R4,=$R5
        048B  0054
        048C  0055
001021  048D  0703                   BIOT  >$+2+$AF
001022  048E  F380 11CB              LNJ   $B7,<ERROR          I/O WAS NAK'ED
001023  0490  C800 13BE              LDR   $R4,<CONT3          FORM I/O CONTROL WORD
001024  0492  C380 10EC              LNJ   $B4,<CGSCH
001025  0494  8070 0037              IO    =55,=$R4
        0496  0054
001026  0497  0703                   BIOT  >$+2+$AF
001027  0498  F380 11CB              LNJ   $B7,<ERROR
001028                       *
001029  049A  C380 10F0              LNJ   $B4,<CHCT           DO CHANNEL CONTROL
001030  049C  0F82                   B     >$+2
001031  049D  0400                   DC    Z'0400'             BLOCK WRITE
001032                       *
001033  049E  C380 117D              LNJ   $B4,<DLAY           TIME DELAY
001034                       *
001035  04A0  C380 10DA              LNJ   $B4,<INXT           INPUT NEXT STATUS
001036                       *
001037  04A2  82D5                   LB    =$R5,=Z'0004'       SEE IF STATUS BIT SET
        04A3  0004
001038  04A4  0503                   BBT   >1564K
001039  04A5  F380 11CB              LNJ   $B7,<ERROR          NON EXISTANT RESOURCE BIT NOT SET
001040  04A7  8382           1564K  JMP   $B2                 JUMP OUT OF TEST
001041                       *
001042                       *-----------------------------------------------------------
001043                       *
001044                       *
001045                       *       CHECK OUTPUT BYTE INTO LCT AND
001046                       *        CHECK INPUT BYTE COMMAND
001047                       *
001048                       *
001049                       *
001050  04A8  9870 4520      LCTOB LDR   $R1,=A'E '
001051  04AA  C380 13E4              LNJ   $B4,<PLB            REPORT TEST
001052  04AC  C380 106D              LNJ   $B4,<GENIZ          GIVE GENERAL INITIALIZE
001053                       *
001054  04AE  8753                   CL    =$R3
001055  04AF  C380 105D      LCTB    LNJ   $B4,<FLN            FIND LINE ON
001056  04B1  8382                   JMP   $B2                 NO LINE ON
001057  04B2  BF00 054B              STR   $R3,<TSTCHN         STORE SUBCH.
001058  04B4  3B00 04B7              BEVN  $R3,<NXLCTA         SET TO READ SUBCH.
001059  04B6  88D3                   DEC   =$R3
001060                       *
001061                       *  SET UP TABLE TO BE SENT IN OUTPUT BYTE TO LCT
001062                       *
001063  04B7  C380 04BA      NXLCTA LNJ   $B4,<LCTGRB          FILL LCT AREA WITH A PATTERN
001064  04B9  0FB6                   B     >LCTZ
001065                       *
001066                       *  DATA IS ASCENDING STARTING WITH ONE
001067                       *
001068  04BA  5CC0           LCTGRB LDV   $R5,=-64             64 ENTRIES IN TABLE
001069  04BB  8751                   CL    =$R1                R1 TRACKS ADDRESS
001070  04BC  A870 0100              LDR   $R2,=X'100'         R2 TRACKS DATA
001071  04BE  C852           NXLCT   LDR   $R4,=$R2
001072  04BF  C451                   OR    $R4,=$R1            FORM DATA AND ADDRESS
001073  04C0  CF10 2400              STR   $R4,<SDB.$R1
001074  04C2  8AD1                   INC   =$R1
001075  04C3  AA70 0100              ADD   $R2,=X'100'         FORM NEXT DATA WORD
001076  04C5  5760 04BE              BINC  $R5,<NXLCT          DO NEXT VALUE
001077                       * NOW DUMMY OUT CONTROL CHARACTERS
001078  04C7  C870 0100              LDR   $R4,=Z'0100'
001079  04C9  8751                   CL    =$R1
001080  04CA  A810 04DD      NXLCT1  LDR   $R2,<NONO.$R1       GET FORBIDDEN CHARACTER
001081  04CC  8AD1                   INC   =$R1
001082  04CD  2800 04D2              BLZ   $R2,<GRB1           BRANCH IF END OF LIST
001083  04CF  CF20 2400              STR   $R4,<SDB.$R2
001084  04D1  0FF9                   B     >NXLCT1
001085                       *
001086                       * SET DATA BUFFER POINTERS TO NON ZERO
001087                       *
001088  04D2  9870 8005      GRB1    LDR   $R1,=Z'8005'
001089  04D4  9F00 2405              STR   $R1,<SDB+5
001090  04D6  9870 4025              LDR   $R1,=Z'4025'
001091  04D8  9F00 2425              STR   $R1,<SDB+X'25'
001092  04DA  8700 2440              CL    <SDB+64
001093  04DC  8384                   JMP   $B4                 GET OUT
001094                       *
001095                       * TABLE OF CONTROL LOCATIONS
001096                       *
001097  04DD  0008           NONO    DC    X'8'                CHANNEL COMMAND REC
001098  04DE  0009                   DC    X'9'                CHANNEL CONTROL REC
001099  04DF  0028                   DC    X'28'               CHAN COMMAND, TRAN
001100  04E0  0029                   DC    X'29'               CHAN CONTROL, TRAN
001101  04E1  000C                   DC    X'C'                IRUPT LEVEL, REC
001102  04E2  000D                   DC    X'D'
001103  04E3  002C                   DC    X'2C'               IRUPT LEVEL, TRAN
001104  04E4  002D                   DC    X'2D'
001105  04E5  0020                   DC    X'20'
001106  04E6  0021                   DC    X'21'
001107  04E7  0000                   DC    X'0'
001108  04E8  0001                   DC    X'1'
001109  04E9  0014                   DC    X'14'               LR CONTROL
001110  04EA  0015                   DC    X'15'
001111  04EB  0036                   DC    X'36'
001112  04EC  0037                   DC    X'37'
001113  04ED  FFFF                   DC    -X'1'               END OF LIST
001114  04EE  0000                   RESV  1,0                 ROOM FOR EXPANSION
001115                       *
001116                       *
001117  04EF  8700 2440      LCTZ    CL    <SDB+64             MARK END OF LIST
001118  04F1  C870 0710              LDR   $R4,=Z'0710'
001119  04F3  CF00 2410              STR   $R4,<SDB+16         SPECIAL DATA FOR STATUS
001120  04F5  C870 1011              LDR   $R4,=Z'1011'
001121  04F7  CF00 2411              STR   $R4,<SDB+17
001122  04F9  B380 0F7D              LNJ   $B3,<SETLCT         SEND LCT BYTES OUT ONE AT A TIME
001123  04FB  2400                   DC    <SDB
001124                       *
001125                       * NOW READ IN  DATA USING INPUT BYTE INSTRUCTION AND COMPARE
001126                       *
001127  04FC  BB80 2440              LAB   $B3,<SDB+64
```

```
001128  04FE  2CC0                    LDV   $R2,=-64
001129                          *
001130  04FF  9823              RBYTES LDR   $R1,$B3,$R2          GET SHOULD BE BYTE
001131  0500  9F56                     STR   $R1,=$R6
001132  0501  1008                     SOL   $R1,8               ALIGN ADDRESS
001133  0502  1900  0510               BEZ   $R1,<RBYTE1         DON'T READ LOC 0
001134  0504  9470  0037               OR    $R1,=X'37'          OR IN BYTE DIRECTOR
001135  0506  C380  0DC3               LNJ   $B4,<OUTLC1         OUTPUT BYTE
001136  0508  C380  1121               LNJ   $B4,<INBYTE         INPUT BYTE TO R5
001137  050A  E570  FF00               AND   $R6,=Z'FF00'        STRIP TO DATA
001138  050C  E955                     CMR   $R6,=$R5
001139  050D  0903                     BE    >RBYTE1
001140  050E  F380  11CB               LNJ   $B7,<ERROR          WRONG DATA INPUT BY INPUT BYTE
001141  0510  2780  04FF        RBYTE1 BINC  $R2,<RBYTES
001142                          *
001143  0512  2CC0                     LDV   $R2,=-64
001144  0513  BB80  2440               LAB   $B3,<SDB+64         SHIFT DATA OVER SO IT
001145  0515  C823              LCTC   LDR   $R4,$B3,$R2         CAN BE PACKED USING PKRIN
001146  0516  4048                     SOR   $R4,8
001147  0517  CF23                     STR   $R4,$B3,$R2
001148  0518  27FD                     BINC  $R2,>LCTC
001149                          *
001150  0519  5C20                     LDV   $R5,=32
001151  051A  C380  112F               LNJ   $B4,<PKRIN          STRIP AND PACK SEND BLOCK TO LOOK
001152  051C  2400                     DC    <SDB                LIKE LCT READ BACK
001153  051D  2500                     DC    <SDB+X'100'
001154                          *
001155  051E  C800  054B               LDR   $R4,<TSTCHN         CALULATE LCT RAM ADDRESS
001156  0520  4B84                     BUDD  $R4,>LCTE
001157  0521  4F20                     MLV   $R4,=X'20'
001158  0522  CF00  0528               STR   $R4,<LCTA
001159                          *
001160  0524  9380  1009        LCTE   LNJ   $B1,<RDATA          READ BACK LCT AREA
001161  0526  2800                     DC    <RTB
001162  0527  0040                     DC    64
001163  0528  0000              LCTA   DC    0                   RAM ADDRESS
001164  0529  0000                     DC    0                   START READ ON EVEN BYTE
001165                          *
001166                          *
001167                          *
001168                          * COMPARE DATA
001169                          *
001170  052A  8751                     CL    =$R1
001171  052B  4CE0                     LDV   $R4,=-32            COUNTER
001172                          *
001173  052C  9B80  2500               LAB   $B1,<SDB+X'100'     GET ADDRESS OF SHOULD BE BUFFER
001174                          *
001175  052E  D810  2800        LCTX   LDR   $R5,<RTB,$R1        GET IS DATA
001176  0530  E85D                     LDR   $R6,$B1,+$R1        GET SHOULD BE DATA
001177  0531  E970  0101               CMR   $R6,=X'0101'        CHECK FOR NULL FLAG
001178  0533  0906                     BE    >LCTW               BYPASS COMPARISON IF CONTROL CHAR
001179  0534  F851                     LDR   $R7,=$R1            DISPLAY WORD
001180  0535  D956                     CMR   $R5,=$R6
001181  0536  0903                     BE    >$+2+$AF            BRANCH IF GOOD
001182  0537  F380  11CB               LNJ   $B7,<ERROR          DATA ERROR
001183  0539  4780  052E        LCTW   BINC  $R4,<LCTX           DO NEXT
001184                          *
001185                          *EECHECK TO SEE IF LCT STATUS IS CLEARED
001186                          *                  (DONE FOR RECEIVE CHANNELS ONLY)
001187                          *
001188  053B  3B80  0544               BUDD  $R3,<NOER5
001189  053D  C380  0836               LNJ   $B4,<ILCTST         READ LCT STATUS
001190                          *
001191                          *
001192  053F  8756                     CL    =$R6                LOAD $R6 WITH 'SHOULD BE'
001193  0540  D956                     CMR   $R5,=$R6
001194  0541  0903                     BE    >$+2+$AF
001195                          *
001196  0542  F380  11CB               LNJ   $B7,<ERROR          LCT STATUS NOT CLEARED
001197                          *
001198  0544  B800  054B        NOER5  LDR   $R3,<TSTCHN         GET HIGHEST SUBCH.
001199  0546  8AD3                     INC   =$R3
001200  0547  3D10                     CMV   $R3,=16
001201  0548  0981  FF66               BNE   LCTB
001202  054A  8382                     JMP   $B2
001203  054B  0000              TSTCHN DC    0
001204                          *
001205                          *-----------------------------------------------------------------
001206                          *
001207                          *
001208                          * SOFT-INITIALIZE TEST
001209                          *
001210  054C  9870  5349        SITEST LDR   $R1,=A'SI'          REPORT TEST
001211  054E  C380  13E4               LNJ   $B4,<PLB
001212                          *
001213  0550  C380  106D        SIT1   LNJ   $B4,<GENIIZ         INITIALIZE
001214                          *
001215  0552  8753                     CL    =$R3
001216  0553  C380  105D               LNJ   $B4,<FLN            FIND LINE NUMBER
001217  0555  8382                     JMP   $B2                 NO LINES
001218                          *
001219                          * FILL WRITE BUFFER WITH DATA (8408)
001220                          *
001221                                  CALL  ZV$F,SDB,CCITT,BUFRNG
         0556  FBC0  0003
         0558  D380  0000        X
         055A  0F80
         055B  2400
         055C  1321
         055D  1340
001222                          *
001223                          * WRITE DATA OUT
001224  055E  BF00  1347        SIT9   STR   $R3,<IPR
001225  0560  3B80  0563               BUDD  $R3,<SIT4
001226  0562  8AD3                     INC   =$R3                GET WRITE CHANNEL
001227                          *
001228  0563  9380  0FDA        SIT4   LNJ   $B1,<SDATA
001229  0565  2400                     DC    <SDB                FROM HERE
001230  0566  0400                     DC    X'400'              RANGE
001231  0567  0300                     DC    X'300'              START AT HEX 300
001232  0568  8000                     DC    Z'8000'             EVEN BYTE BOUNDRY
001233                          *
001234                          * WRITE OUT CHANNEL PROGRAM
```

```
001235                                  *
001236  0569  9380 0FDA                         LNJ     $B1,<SDATA              SEND DATA
001237  056B  05B7                             DC      <ITEST
001238  056C  0004                             DC      (ITEST1-ITEST)*2        RANGE
001239  056D  0200                             DC      X'200'                  RAM ADDRESS
001240  056E  0000                             DC      0                       EVEN BYTE
001241                                  *
001242                                  * LOAD DUMMY INFO TO LCT 8, X'28'
001243                                  *
001244  056F  B380 0F7D                         LNJ     $B3,<SETLCT
001245  0571  05B2                             DC      <IZLCT
001246                                  *
001247                                  * READ LCT 8,9,X'26',X'29'
001248                                  *
001249  0572  B800 1347                         LDR     $R3,<IPR                GET TESTED CHANNEL
001250  0574  B380 063E                         LNJ     $B3,<STRT10             START CHANNEL PROGRAM
001251  0576  C380 117D                         LNJ     $B4,<DLAY               DELAY 25 MS
001252  0578  9B80 05B9                 SIT2    LAB     $B1,<TSVLU
001253  057A  8752                             CL      =$R2
001254  057B  986D                     SIT6    LDR     $R1,$B1,+$R2            GET BYTE ADDRESS
001255  057C  C380 0DC3                         LNJ     $B4,<OUTLCT             OUTPUT II
001256  057E  C380 1121                         LNJ     $B4,<INBYTE             INPUT BYTE
001257  0580  5903                             BEZ     $R5,>SIT5               BRANCH IF CLEARED
001258  0581  F380 11CB                         LNJ     $B7,<ERROR              LCT NOT CLEARED
001259  0583  2D04                     SIT5    CMV     $R2,=4                  CHECK FOR LAST IN TABLE
001260  0584  09F7                             BNE     >SIT6
001261                                  *
001262                                  * TEST SOFTWARE SOFT INITIALIZE
001263                                  *
001264                                  *
001265  0585  C380 0DDC                         LNJ     $B4,<RPVLU              READ P VALUE
001266  0587  E670 0202                         LDR     $R6,=X'202'             WHAT IT SHOULD BE
001267  0589  D956                             CMR     $R5,=$R6
001268  058A  0900 058E                         BE      <SIT7
001269  058C  F380 11CB                         LNJ     $B7,<ERROR              CCP SOFT INIT FAILURE
001270                                  *
001271  058E  B800 1347                 SIT7    LDR     $R3,<IPR
001272  0590  3B00 0593                         BEVN    $R3,<SIT11
001273  0592  88D3                             DEC     =$R3
001274                                  *
001275  0593  9380 1009                 SIT11   LNJ     $B1,<RDATA              READ DATA
001276  0595  2800                             DC      <RTB                    TO HERE
001277  0596  0400                             DC      X'400'                  RANGE
001278  0597  0300                             DC      X'300'                  RAM ADDRESS
001279  0598  8000                             DC      Z'8000'                 EVEN BYTE, 250 MS DELAY
001280                                  *
001281  0599  8700 135E                         CL      <MASK
001282  059B  9870 0200                         LDR     $R1,=X'200'
001283  059D  9F00 1354                         STR     $R1,<RANGEW             RANGE IN WORDS
001284  059F  9380 0458                         LNJ     $B1,<CRWB               COMPARE DATA
001285  05A1  C380 10F0                 SIT10   LNJ     $B4,<CHCT               CHANNEL CONTROL
001286  05A3  0F82                             B       >$+2
001287  05A4  0100                             DC      X'100'                  CCB LIST RESET
001288                                  *
001289  05A5  8AD3                             INC     =$R3                    GET WRITE CHANNEL
001290  05A6  C380 10F0                         LNJ     $B4,<CHCT               CHANNEL CONTROL
001291  05A8  0F82                             B       >$+2
001292  05A9  0100                             DC      X'100'                  CCB LIST RESET
001293                                  *
001294                                  *
001295  05AA  B800 1347                         LDR     $R3,<IPR                GET TESTED CHANNEL
001296  05AC  8AD3                             INC     =$R3                    BUMP FOR NEXT CHANNEL
001297  05AD  C380 105D                         LNJ     $B4,<FLN                FIND NEXT LINE NUMBER
001298  05AF  8382                             JMP     $B2                     NO MORE LINES, DONE
001299  05B0  0F80 055E                         B       <SIT9
001300                                  *
001301                                  *
001302                                  * LCT FOR THIS TEST
001303  05B2  4008                      IZLCT   DC      Z'4008'                 DATA SET SCAN
001304  05B3  4028                             DC      Z'4028'
001305  05B4  0206                      IZL     DC      Z'0206'                 P, MSB RCV
001306  05B5  0226                             DC      Z'0226'                 P MSB, XMIT
001307  05B6  0000                             DC      0
001308                                  *
001309                                  * CHANNEL PROGRAM
001310                                  *
001311  05B7  0009                      ITEST   DC      Z'0009'                 NOP,INZ
001312  05B8  0001                              DC      Z'0001'                 NOP,WAIT
001313        05B9                      ITEST1  EQU     $
001314                                  *
001315                                  * TEST ADDRESSES
001316                                  *
001317  05B9  0837                      TSVLU   DC      Z'0837'
001318  05BA  0937                              DC      Z'0937'
001319  05BB  2837                              DC      Z'2837'
001320  05BC  2937                              DC      Z'2937'
001321                                  *
001322                                  * CHANNEL PROGRAMS FOR NEXT TEST
001323                                  *
001324  05BD  0001                      PROG1   DC      Z'0001'                 NOP,WAIT
001325  05BE  0001                              DC      Z'0001'                 NOP, WAIT
001326  05BF  E0FE                              DC      Z'E0FE'                 B -2
001327                                  *
001328  05C0  5029                      PROG2   DC      Z'5029'                 LD 29
001329  05C1  93FB                              DC      Z'93FB'                 AND =X'FB' (START I/O BIT)
001330                                  * !! THE FOLLOWING STORE IS MODIFIED TO A NOP IF LESS THEN REV 11 MLCP FW
001331  05C2  5129                      PROG3   DC      Z'5129'                 ST 29, PUTS BACK, STRIPPED
001332  05C3  0000                              DC      Z'0000'                 NOP
001333  05C4  0000                              DC      X'0'
001334  05C5  0000                              DC      X'0'                    NOP, NOP
001335  05C6  5027                              DC      Z'5027'                 LD 27
001336  05C7  E202                              DC      Z'E202'                 BZT +2
001337  05C8  0100                              DC      Z'0100'                 WAIT, NOP
001338  05C9  5029                              DC      Z'5029'                 LD 29
001339  05CA  93FB                              DC      Z'93FB'                 AND =FB (START I/O BIT)
001340  05CB  5129                              DC      Z'5129'                 ST 29, PUTS BACK STRIPPED
001341  05CC  0000                              DC      0
001342                                  *
001343                                  *-----------------------------------------------------------
001344                                  *
001345                                  *
001346                                  *          CHANNEL PROGRAM TEST (NOP PROGRAM)
001347                                  *
```

```
001348                          *
001349  05CD  9870 4620         CHPGT   LDR     $R1,=A'F '
001350  05CF  C380 13E4                 LNJ     $B4,<PLB                REPORT TEST
001351  05D1  8753                      CL      =$R3                    INITIALIZE CHANNEL NUMBER
001352  05D2  C380 106D                 LNJ     $B4,<GENITZ             GENERAL INITIALIZE
001353                          *
001354                          *  MODIFY CHANNEL PROGRAM IF REV 11 OR HIGHER FIRMWARE
001355                          *
001356  05D4  9800 133E                 LDR     $R1,<FW-REV             GET FW REV
001357  05D6  1D0B                      CMV     $R1,=11
001358  05D7  0283                      BGE     >CHPGT3                 B = REV 11 OR HIGHER
001359  05D8  8700 05C2                 CL      <PROG3
001360  05DA  C380 105D         CHPGT3  LNJ     $B4,<FLN                FIND LINE ON
001361  05DC  8382                      JMP     $B2                     NO LINE ON JUMP OUT OF TEST
001362  05DD  3B80 05E0                 BOUD    $R3,<CHPGT2             GET WRITE SUBCH.
001363  05DF  8AD3                      INC     =$R3
001364  05E0  8751              CHPGT2  CL      =$R1                    CLEAR INDEX FOR OUTPUT ADDRESS
001365  05E1  BF00 1347                 STR     $R3,<TPR                STORE SUBCH.
001366                          *
001367                          *     RAM SHOULD BE CLEARED TO 0 (NOPS) BY LAST INITIALIZE
001368                          *
001369  05E3  9380 0FDA                 LNJ     $B1,<SDATA              SEND DATA
001370  05E5  05C0                      DC      <PROG2
001371  05E6  001A                      DC      26                      RANGE IN BYTES
001372  05E7  0218                      DC      X'218'                  RAM ADDRESS
001373  05E8  0000                      DC      0                       EVEN BYTE IN CPU
001374                          *
001375  05E9  9380 0FDA                 LNJ     $B1,<SDATA              WRITE CHANNEL PROGRAM
001376  05EB  05BD                      DC      <PROG1
001377  05EC  0006                      DC      X'6'                    RANGE
001378  05ED  0DFE                      DC      X'DFE'
001379  05EE  0000                      DC      0                       EVEN BYTE BOUNDRY
001380                          *
001381                          *
001382  05EF  B380 0F7D                 LNJ     $B3,<SETLCT             SEND LCT OUT FOR THIS CHANNEL
001383  05F1  064D                      DC      <CP14
001384                          *
001385  05F2  C380 117D                 LNJ     $B4,<DLAY
001386  05F4  B380 063E                 LNJ     $B3,<STRTIO             START IO
001387                          *
001388  05F6  C380 117D                 LNJ     $B4,<DLAY               TIME DELAY
001389                          *
001390  05F8  C380 0DDC                 LNJ     $B4,<RPVLU
001391                          *
001392  05FA  E870 0229                 LDR     $R6,=Z'0229'
001393  05FC  CF00 1351                 STR     $R4,<IRG
001394  05FE  D956                      CMR     $R5,=$R6
001395  05FF  0903                      BE      >$+2+$AF
001396  0600  F380 11CB                 LNJ     $B7,<ERROR              PAUSE FUNCTION NOT WORKING
001397                          *
001398                          * NOW DISABLE PAUSE
001399                          *
001400  0602  B380 0F7D                 LNJ     $B3,<SETLCT             OUTPUT BYTES
001401  0604  0649                      DC      <CP12                   FROM HERE
001402                          *
001403                          * TEST THAT STOP IO COMMAND WILL INTERRUPT CHANNEL PROGRAM.
001404                          *
001405  0605  B380 063E                 LNJ     $B3,<STRTIO             GIVE START IO
001406  0607  1CB0                      LDV     $R1,=-80
001407  0608  0F03             NP4     NOP     >$+2+$AF
001408  0609  1780 0608                 BINC    $R1,<NP4                SHORT DELAY FOR CHAN PROG TO START.
001409  060B  C380 10F0                 LNJ     $B4,<CHCT
001410  060D  0F82                      B       >$+2
001411  060E  2000                      DC      Z'2000'                 STOP IO
001412                          *
001413  060F  C380 117D                 LNJ     $B4,<DLAY               DELAY 25MS
001414  0611  C380 0DDC                 LNJ     $B4,<RPVLU              READ P TO TEMP
001415                          *
001416  0613  D956                      CMR     $R5,=$R6                R6 STILL HAS X'229'
001417  0614  0303                      BG      >NP5
001418  0615  F380 11CB                 LNJ     $B7,<ERROR              PROGRAM DIDN'T START AGAIN
001419                          *
001420  0617  E870 0E00         NP5     LDR     $R6,=X'E00'
001421  0519  D956                      CMR     $R5,=$R6
001422  061A  0203                      BL      >NP6
001423  061B  F380 11CB                 LNJ     $B7,<ERROR              ERROR, STOP I/O DIDN'T STOP  CCP
001424                          * START CHANNEL PROGRAM AGAIN
001425                          *
001426                          * RESET P TO START AT LOC WHICH RESETS START I/O BIT
001427                          *
001428  061D  B380 0F7D         NP6     LNJ     $B3,<SETLCT
001429  061F  064C                      DC      <CP13
001430  0620  B380 063E                 LNJ     $B3,<STRTIO             GIVE START IO
001431  0622  1CB0                      LDV     $R1,=-80
001432  0623  0F03             NP7     NOP     >$+2+$AF
001433  0624  1780 0623                 BINC    $R1,<NP7                DELAY TO LET CHANNEL PROGRAM START
001434                          *
001435                          * GIVE STOP I/O FOR OTHER CHANNEL.  SHOULDN'T STOP PROGRAM.
001436                          *
001437  0626  8BD3                      DEC     =$R3
001438  0627  C380 10F0                 LNJ     $B4,<CHCT               DO CHANNEL CONTROL
001439  0629  0F82                      B       >$+2
001440  062A  2000                      DC      Z'2000'                 STOP IO
001441                          *
001442                          *
001443                          * GIVE OUTPUT BYTE COMMANDS TO MAKE SURE THEY DON'T STOP CHANNEL PROG.
001444                          *
001445  062B  8AD3                      INC     =$R3                    GET BACK CHANNEL
001446  062C  B380 0F7D                 LNJ     $B3,<SETLCT             OUTPUT LCT AGAIN
001447  062E  0649                      DC      <CP12                   SAME AS BEFORE
001448                          *
001449                          * NOW GIVE INPUT DATA SET STATUS TO CHECK IT WONT TERMINATE CHAN
001450                          * PROGRAM
001451                          *
001452  062F  C380 0E77                 LNJ     $B4,<DSSTA
001453  0631  C380 117D                 LNJ     $B4,<DLAY               DELAY 25 MS
001454                          *
001455  0633  C380 0DDC                 LNJ     $B4,<RPVLU              READ P TO TEMP
001456                          *
001457                          *
001458  0635  E870 0E00                 LDR     $R6,=Z'0E00'            FORM 'SHOULD BE' P COUNTER
001459  0637  CF00 1351                 STR     $R4,<IRG                STORE FOR ERROR
001460  0639  D956                      CMR     $R5,=$R6                COMPARE P COUNTER
```

```
001461  063A  0903                        BE      >$+2+$AF
001462  063B  F380 11CB                    LNJ     $B7,<ERROR          P COUNT WRONG, BAD PROGRAM EXICUTION
001463  063D  8382                        JMP     $B2                 DONE WITH TEST
001464                              *
001465                              * START IO
001466                              *
001467  063E  C800 13C2             STRTIO LDR     $R4,<CONT7          GET FUNC CODE
001468  0640  C380 10EC                    LNJ     $B4,<CGSCH
001469  0642  8070 4000                    IO      =Z'4000',=$R4
      0644  0054
001470  0645  0703                        BIOT    >$+2+$AF
001471  0646  F380 11CB                    LNJ     $B7,<ERROR          COMMAND WAS NAK'ED
001472  0648  8383                        JMP     $B3
001473                              *
001474  0649  0105                  CPT2   DC      Z'0105'             RCV PAUSE DISABLE
001475  064A  0125                         DC      Z'0125'             XMIT PAUSE DISABLE
001476  064B  0000                         DC      0
001477                              *
001478  064C  2927                  CPT5   DC      X'2927'             RESTART AT 229
001479  064D  0226                  CPT4   DC      X'0226'             P,MSB,XMIT
001480  064E  0000                         DC      0                   END OF LIST
001481                              *
001482                              *-------------------------------------------------------------
001483                              *
001484                              *           INSTRUCTION TEST
001485                              *
001486                              *
001487  064F  9870 4720             INSIT  LDR     $R1,=A'G '
001488  0651  C380 13E4                    LNJ     $B4,<PLD
001489  0653  8753                        CL      =$R3                INITIALIZE CHANNEL NUMBER
001490                              *
001491  0654  C380 105D             INSLUP LNJ     $B4,<FLN            FIND LINE ON
001492  0656  8382                        JMP     $B2                 NO LINE ON JUMP OUT OF TEST
001493  0657  BF00 1347                    STR     $R3,<IPR            STORE SUBCH. NUMBER
001494  0659  BF00 1349                    STR     $R3,<IEMP1
001495  065B  C380 106D                    LNJ     $B4,<GENITZ         MLCC GENERAL INITIALIZE
001496  065D  3B82                        BODD    $R3,>INS2           GET WRITE SUBCH.
001497  065E  8AD3                        INC     =$R3
001498  065F  8751                 INS2   CL      =$R1                CLEAR INDEX FOR OUTPUT ADDRESS
001499  0660  C870 02B1                    LDR     $R4,=X'2B1'         LOAD MASK1 WITH EXPECTED
001500  0662  CF00 135F                    STR     $R4,<MASK1          P COUNTER
001501                              *
001502                              *
001503                              *
001504                              *        PREPARE AND LOAD RAM INSTRUCTIONS
001505                              *
001506                              *
001507  0664  9380 0FDA                    LNJ     $B1,<SDATA
001508  0666  1202                        DC      <INSTR1
001509  0667  007A                        DC      (INSTR2-INSTR1)*2
001510  0668  0200                        DC      X'200'              RAM ADDRESS
001511  0669  0000                        DC      0                   CPU HAS RIGHT BYTE START
001512                              *
001513                              *
001514  066A  9380 0FDA                    LNJ     $B1,<SDATA
001515  066C  123F                        DC      <INSTR2
001516  066D  0038                        DC      (INSTR3-INSTR2)*2
001517  066E  027A                        DC      X'27A'              RAM ADDRESS
001518  066F  0000                        DC      0                   EVEN BYTE START
001519                              *
001520                              *
001521  0670  9380 0FDA                    LNJ     $B1,<SDATA
001522  0672  125B                        DC      <INSTR3
001523  0673  002E                        DC      (INSTR4-INSTR3)*2
001524  0674  0306                        DC      X'306'              RAM ADDRESS
001525  0675  0000                        DC      0                   EVEN BYTE START
001526                              *
001527  0676  9380 0FDA                    LNJ     $B1,<SDATA          SEND DATA
001528  0678  1272                        DC      <INSTR4
001529  0679  0006                        DC      (INSTR5-INSTR4)*2   RANGE
001530  067A  05FD                        DC      X'5FD'              RAM ADDRESS
001531  067B  0000                        DC      0                   EVEN BYTE IN CPU
001532                              *
001533  067C  9380 0FDA                    LNJ     $B1,<SDATA          SEND DATA
001534  067E  1275                        DC      <INSTR5
001535  067F  0006                        DC      (INSTR6-INSTR5)*2   RANGE
001536  0680  0AFE                        DC      X'AFE'              RAM ADDRESS
001537  0681  0000                        DC      0                   EVEN BYTE IN CPU
001538                              *
001539  0682  9380 0FDA                    LNJ     $B1,<SDATA          SEND DATA
001540  0684  1278                        DC      <INSTR6
001541  0685  0008                        DC      (TRANM1-INSTR6)*2   RANGE
001542  0686  0DFB                        DC      X'DFB'              RAM ADDRESS
001543  0687  0000                        DC      0                   EVEN BYTE IN CPU
001544                              *
001545                              *
001546                              * SEND OUT LOOKUP TABLE FOR INSTRUCTION TEST
001547                              *
001548  0688  9380 0FDA                    LNJ     $B1,<SDATA          SEND PROGRAM
001549  068A  06BB                        DC      <TALU
001550  068B  0004                        DC      4                   RANGE OF 4 BYTES
001551  068C  0402                        DC      X'402'              RAM ADDRESS
001552  068D  0000                        DC      0                   STARTS ON EVEN BOUNDRY
001553                              *
001554                              *
001555                              *
001556  068E  B380 0F7D                    LNJ     $B3,<SETLCT         SEND LCT OUT
001557  0690  06B0                        DC      <LCTTAB
001558  0691  B380 063E                    LNJ     $B3,<STRTIO         START IO
001559                              *
001560                              *
001561  0693  C380 117D                    LNJ     $B4,<DLAY           TIME DELAY
001562                              *
001563  0695  C853                        LDR     $R4,=$R3            FIND RAM ADDRESS TO READ P COUNTER
001564  0696  4F20                        MLV     $R4,=X'20'
001565  0697  4E06                        ADV     $R4,=6
001566  0698  CF00 069F                    STR     $R4,<INS1
001567  069A  88D3                        DEC     =$R3                GET READ SUBCH.
001568                              *
001569  069B  9380 1009                    LNJ     $B1,<RDATA          READ P COUNTER BACK
001570  069D  1354                        DC      <RANGEW
001571  069E  0002                        DC      2
001572  069F  0000                 INS1   DC      0
```

```
001573  06A0  0000                        DC      0
001574                            *
001575                            *
001576  06A1  D800 1354                    LDR     $R5,<RANGEW
001577  06A3  D570 0FFF                    AND     $R5,=X'FFF'
001578  06A5  E800 135F                    LDR     $R6,<MASK1          GET SHOULD BE DATA
001579  06A7  D956                        CMR     $R5,=$R6            SEE IF P COUNTER RIGHT?
001580  06A8  0903                        BE      >NOINF              IF NOT EQUAL HALT
001581  06A9  F380 11CB                    LNJ     $B7,<ERROR          INSTRUCTION TEST ERROR $R5 CONTAINS PCOUNTER
001582  06AB  8800 1347           NOINF   LDR     $R3,<TPR            GET BACK TESTED CHANNEL
001583  06AD  8AD3                        INC     =$R3                BUMP CHANNEL NUMBER
001584  06AE  0F81 FFA5                    B       INSLOP              DO FOR NEXT CHANNEL
001585                            *
001586                            * LCT TABLE FOR THIS TEST
001587                            *
001588  06B0  0226                LCTTAB  DC      Z'0226'             LCT CCP POINTER (MSB)
001589  06B1  7A27                        DC      Z'7A27'             CCP POINTER (LSB)
001590  06B2  FF03                        DC      Z'FF03'             ADDRESS TESTING VALUE
001591  06B3  AA15                        DC      Z'AA15'             DITTO
001592  06B4  552C                        DC      Z'552C'             DITTO
001593  06B5  AA3F                        DC      Z'AA3F'             DITTO
001594  06B6  0437                        DC      Z'0437'             TLU POINTER, MSB
001595  06B7  0238                        DC      Z'0238'             TLU POINTER, LSB
001596  06B8  0105                        DC      Z'0105'
001597  06B9  0125                        DC      Z'0125'
001598  06BA  0000                        DC      0                   MARKS END OF TABLE
001599                            *
001600                            * TABLE FOR TABLE LOOKUP TEST
001601                            *
001602  06BB  031A                TALU    DC      X'031A'
001603  06BC  8100                        DC      Z'8100'
001604                            *-------------------------------------------------------------
001605                            *
001606                            * OUTPUT INTERRUPT CONTROL TEST
001607                            *
001608  06BD  9870 4820           INCONT  LDR     $R1,=A'H '
001609  06BF  C380 13E4                    LNJ     $B4,<PLB            REPORT TEST
001610  06C1  8753                        CL      =$R3                R3 TRACKS CHANNEL
001611  06C2  C380 105D           ICON3   LNJ     $B4,<FLN            FIND ACTIVE LINE NUMBER
001612  06C4  8382                        JMP     $B2                 NO MORE LINES
001613  06C5  8C56                        STS     =$R6                GET S REG
001614  06C6  E570 0FC0                    AND     $R6,=X'FC0'         STRIP TO RETURN CHANNEL NUMBER
001615  06C8  6E3F                        ADV     $R6,=63             SET FOR LEVEL 63
001616  06C9  C380 106D                    LNJ     $B4,<GENITZ         GENERAL INITIALIZE
001617  06CB  C800 13C5                    LDR     $R4,<CONT11         OUTPUT INTERRUPT CONTROL  FUNCTION
001618  06CD  C380 10EC                    LNJ     $B4,<CGSCH          MODIFY FOR CHANNEL
001619  06CF  8056                        IO      =$R6,=$R4           OUTPUT INTERRUPT CONTROL
        06D0  0054
001620  06D1  0703                        BIOT    >$+2+$AF
001621  06D2  F380 11CB                    LNJ     $B7,<ERROR          COMMAND WAS NAK'ED
001622                            *
001623  06D4  C853                        LDR     $R4,=$R3
001624  06D5  4F20                        MLV     $R4,=X'20'          OFFSET FOR CHANNEL
001625  06D6  4E0C                        ADV     $R4,=X'C'           OFFSET FOR RETURN CHANNEL
001626  06D7  CF00 06E2                    STR     $R4,<ICON2
001627  06D9  8F00 1347                    STR     $R3,<TPR
001628  06DB  3B00 06DE                    BEVN    $R3,<ICON1
001629  06DD  88D3                        DEC     =$R3                GET RCV CHANNEL FOR BLOCK READ
001630                            *
001631  06DE  9380 1009           ICON1   LNJ     $B1,<RDATA
001632  06E0  1348                        DC      <TEMP               READ TO HERE
001633  06E1  0002                        DC      2                   2 BYTES OF DATA
001634  06E2  0000                ICON2   DC      0                   RAM  ADDRESS
001635  06E3  0000                        DC      0                   EVEN BYTE ADDRESS
001636                            *
001637  06E4  8800 1347                    LDR     $R3,<TPR            RESTORE R3
001638  06E6  D800 1348                    LDR     $R5,<TEMP
001639  06E8  D956                        CMR     $R5,=$R6            CHECK IS VS SB
001640  06E9  0903                        BE      >$+2+$AF
001641  06EA  F380 11CB                    LNJ     $B7,<ERROR          ERROR, OUTPUT INTERRUPT CONTROL
001642  06EC  8AD3                        INC     =$R3                SET FOR NEXT CHANNEL
001643  06ED  0F81 FFD4                    B       ICON3               DO NEXT CHAN
001644                            *
001645                            *-------------------------------------------------------------
001646                            *
001647                            * INITIALIZE AND CHANNEL RESET TEST
001648                            *
001649                            *   FIRST GIVE GEN INITIALIZE + CHECK IT CLEARS LCT AREA
001650                            *
001651  06EF  9870 4A20           IZTST   LDR     $R1,=A'J '
001652  06F1  C380 13E4                    LNJ     $B4,<PLB            REPORT TEST
001653  06F3  8753                        CL      =$R3                R3 TRACKS CHANNEL
001654  06F4  C380 105D           IZTST6  LNJ     $B4,<FLN            FIND ACTIVE LINES
001655  06F6  8382                        JMP     $B2                 NO MORE ACTIVE LINES
001656                            *
001657  06F7  C380 04BA                    LNJ     $B4,<LCTGRB         SEND GARBAGE TO LCT AREA
001658  06F9  C800 13BC                    LDR     $R4,<CONT1          IOLD FUNCTION CODE
001659  06FB  C380 10EC                    LNJ     $B4,<CGSCH          MODIFY FOR CHANNEL
001660  06FD  CF52                        STR     $R4,=$R2
001661  06FE  C800 13C3                    LDR     $R4,<CONT9          INITIALIZE FUNCTION CODE
001662  0700  C380 10EC                    LNJ     $B4,<CGSCH
001663  0702  8070 8000                    IO      =Z'8000',=$R4       INITIALIZE
        0704  0054
001664  0705  0703                        BIOT    >$+2+$AF
001665  0706  F380 11CB                    LNJ     $B7,<ERROR          INITIALIZE WAS NAK'ED
001666                            *
001667  0708  81C0 0C3F                    IOLD    TEMP,=$R2,=X'1'     GIVE ANOTHER INSTRUCTION
        070A  0052
        070B  0070 0001
001668  070D  0783                        BIOF    >$+2+$AF
001669  070E  F380 11CB                    LNJ     $B7,<ERROR          SHOULD BE NAK'ED UNTIL INITIALIZE DONE
001670                            *
001671  0710  8070 8000                    IO      =Z'8000',=$R4       GIVE ANOTHER INITIALIZE
        0712  0054
001672  0713  0703                        BIOT    >$+2+$AF
001673  0714  F380 11CB                    LNJ     $B7,<ERROR          INITIALIZE SHOULD  NOT BE NAK'ED
001674                            *
001675  0716  C380 1184                    LNJ     $B4,<DLAYLG         DELAY 225 MS
001676                            * SET DEFAULT VALUE IN READ BUFFER
001677  0718  9870 0100                    LDR     $R1,=X'100'
001678  071A  9F00 1352                    STR     $R1,<RANGE
001679                                    CALL    ZV$F,RTB,DFLT,RANGE
        071C  FBC0 0003
```

```
         071E  D380 0000       X
         0720  0F80
         0721  2800
         0722  133A
         0723  1352
001680                         * READ IN LCT AREA FOR THIS CHANNEL AND CHECK
001681   0724  C380 07BE               LNJ     $B4,<BRLCLR             CHECK LCT IS CLEAR
001682   0726  B800 1347               LDR     $R3,<IPR                RESTORE CHAN NUMBER
001683                         *-----------------------------------------------------------------
001684                         *
001685                         * CHANNEL INITIALIZE TEST
001686                         *
001687                         *   SEND OUT  GARBAGE TO LCT AREA
001688                         *
001689   0728  C380 04BA               LNJ     $B4,<LCTGRB
001690   072A  C870 0100               LDR     $R4,=Z'0100'
001691   072C  CF00 2410               STR     $R4,<SDB+16
001692   072E  CF00 2411               STR     $R4,<SDB+17
001693   0730  CF00 2405               STR     $R4,<SDB+5
001694   0732  CF00 2430               STR     $R4,<SDB+48
001695   0734  CF00 2431               STR     $R4,<SDB+49
001696   0736  CF00 2425               STR     $R4,<SDB+37
001697   0738  B380 0F7D               LNJ     $B3,<SETLCT             SEND OUT LCT BYTES
001698   073A  2400                    DC      <SDB
001699                         *
001700                         * NOW SET UP 4 CCB'S TO INSURE INITIALIZE CLEARS  OLD POINTER
001701   073B  9380 07FA               LNJ     $B1,<SET4
001702                         * ADVANCE INPUT NEXT STAT POINTER A COUPLE OF TIMES
001703   073D  9380 07DF               LNJ     $B1,<ADV2
001704                         * NOW GIVE CHANNEL INITIALIZE
001705   073F  C800 13C2               LDR     $R4,<CONT7              CHANNEL CONTROL
001706   0741  C380 10EC               LNJ     $B4,<CGSCH              MODIFY FOR CHANNEL
001707   0743  8070 8000               IO      =Z'8000',=$R4
         0745  0054
001708   0746  0703                    BIOT    >$+2+$AF
001709   0747  F380 11CB               LNJ     $B7,<ERROR              OUTPUT CHAN CONT NAK'ED
001710   0749  C380 117D               LNJ     $B4,<DLAY               DELAY 25 MS
001711                         * USE OTHER SIDE OF LINE TO BLOCK READ CCB AREA + CHECK
001712   074B  BF00 1347               STR     $R3,<IPR
001713   074D  3F20                    MLV     $R3,=X'20'
001714   074E  BA70 0E00               ADD     $R3,=Z'0E00'
001715   0750  BF00 075A               STR     $R3,<CADD               ADDRESS OF START OF CCB AREA
001716   0752  B800 1347               LDR     $R3,<IPR
001717   0754  8B53                    LBC     =$R3,=Z'0001'           GET OTHER SIDE OF LINE
         0755  0001
001718                         *
001719   0756  9380 1009               LNJ     $B1,<RDATA              BLOCK READ
001720   0758  2800                    DC      <RTB                    RETURN BUFFER
001721   0759  0020                    DC      X'20'                   20 BYTE RANGE
001722   075A  0000             CADD    DC      0                       ADDRESS
001723   075B  0000                    DC      0                       EVEN BYTE BOUNDRY
001724                         *
001725                         * CHECK CONTROL BYTES WERE CLEARED
001726   075C  1CFC                    LDV     $R1,=-4
001727   075D  2C05                    LDV     $R2,=5
001728   075E  D0A0 2800        LUP     LDH     $R5,<RTB.$R2            GET CONTROL BYTE
001729   0760  5903                    BEZ     $R5,>LUP1               BRANCH IF GOOD
001730   0761  F380 11CB               LNJ     $B7,<ERROR              CHAN INIT. DIDN'T RESET DATA POINTERS IN LR 5
001731   0763  2E08             LUP1    ADV     $R2,=8                  SET FOR NEXT CONTROL BYTE
001732   0764  17FA                    BINC    $R1,>LUP                DO FOR NEXT
001733   0765  B800 1347               LDR     $R3,<IPR                GET BACK TESTED SUBCHANNEL
001734                         *
001735                         * CHECK INPUT NEXT STATUS WILL BE NAK'ED AFTER CHANNEL INITIALIZE
001736                         *
001737   0767  C800 13BF               LDR     $R4,<CONT4
001738   0769  C380 10EC               LNJ     $B4,<CGSCH
001739   076B  8055                    IO      =$R5,=$R4               GIVE INPUT NEXT STATUS COMMAND
         076C  0054
001740   076D  0783                    BIOF    >$+2+$AF                SHOULD BE NAK'ED
001741   076E  F380 11CB               LNJ     $B7,<ERROR              COMMAND WASN'T NAK'ED
001742   0770  0F07                    NOP     >ECO2
001743                         *
001744                         * SET UP 4 CCB'S  AGAIN TO CHECK LOAD POINTER WAS RESET
001745   0771  9380 07FA        ECO3    LNJ     $B1,<SET4
001746   0773  9380 07E4               LNJ     $B1,<CHKRNG             CHECK STATUS POINTER IS CORRECT
001747   0775  9380 07DF               LNJ     $B1,<ADV2               ADVANCE POINTER 2
001748                         *
001749                         * TEST RESET CCB LIST COMMAND
001750                         *
001751   0777  B380 0F7D        ECO2    LNJ     $B3,<SETLCT             SET DATA BUFF POINTER'S NON ZERO
001752   0779  07BB                    DC      <DIPT
001753   077A  C800 13C2               LDR     $R4,<CONT7              CHANNEL CONTROL
001754   077C  C380 10EC               LNJ     $B4,<CGSCH              MODIFY FOR  CHANNEL
001755   077E  8070 0100               IO      =Z'0100',=$R4           RESET CCB LIST
         0780  0054
001756   0781  0703                    BIOT    >$+2+$AF
001757   0782  F380 11CB               LNJ     $B7,<ERROR              COMMAND WAS NAK'ED
001758                         * SET UP 4 CCB'S AGAIN TO CHECK IF CCB RESET WORKS
001759                         *
001760   0784  9380 07FA               LNJ     $B1,<SET4               SET UP 4 CCB'S
001761   0786  9380 07E4               LNJ     $B1,<CHKRNG             CHECK INPUT STAT POINTER  IS RESET
001762                         *
001763                         * NOW CHECK CHANNEL LCT AREA WAS CLEARED BY INITIALIZE
001764                         *
001765   0788  C380 10F0               LNJ     $B4,<CHCT
001766   078A  0F82                    B       >$+2
001767   078B  0100                    DC      X'100'                  RESET CCB LIST
001768                         *
001769                         *
001770                         *     SET UP DEFAULT
001771                         *
001772   078C  4C45                    LDV     $R4,=X'45'
001773   078D  CF00 1354               STR     $R4,<RANGEW
001774   078F  9380 044F               LNJ     $B1,<SET
001775   0791  C380 07BE               LNJ     $B4,<BRLCLR             CHECK LCT AREA FOR CHAN IS CLEAR
001776                         *
001777                         * TEST BVBT + BVBF COMMANDS
001778                         *
001779   0793  C380 106D               LNJ     $B4,<GENITZ             GENERAL INITIALIZE
001780                         *
001781   0795  9380 0FDA               LNJ     $B1,<SDATA              SEND CHANNEL PROGRAM
001782   0797  130F                    DC      <VALTST                 ADDRESS
001783   0798  0022                    DC      (ECP-VALTST)*2          RANGE
```

```
001784  0799  0200                    DC      X'200'              RAM ADDRESS
001785  079A  0000                    DC      0                   EVEN CP ADDRESS
001786                          *
001787                          * SET UP RCV, XMIT FOR P START AT 200
001788                          *
001789  079B  B380 0F7D                LNJ     $B3,<SETLCT
001790  079D  05B4                    DC      <1ZL
001791                          *
001792                          * SEND OUT 4 CCB'S WITH VALID BITS
001793  079E  9380 07FA                LNJ     $B1,<SET4
001794                          *
001795  07A0  B380 063E                LNJ     $B3,<STRTIO         START CHANNEL PROGRAM
001796  07A2  C380 117D                LNJ     $B4,<DLAY
001797  07A4  C380 10F0                LNJ     $B4,<CHCT           CHANNEL CONTROL
001798  07A6  0F82                    B       >$+2
001799  07A7  0100                    DC      X'100'              RESET CCB LIST
001800                          *
001801                          * START SECOND HALF OF CHANNEL PROGRAM
001802                          *
001803  07A8  B380 063E                LNJ     $B3,<STRTIO
001804  07AA  C380 117D                LNJ     $B4,<DLAY
001805                          * READ P TO INSURE CORRECT CHAN PROGRAM OPERATION
001806  07AC  C380 10F0                LNJ     $B4,<CHCT           CHANNEL CONTROL
001807  07AE  0F82                    B       >$+2
001808  07AF  0100                    DC      X'100'              RESET CCB LIST
001809  07B0  C380 0DDC                LNJ     $B4,<RPVLU          READ P
001810  07B2  E870 0222                LDR     $R6,=X'222'
001811  07B4  D956                    CMR     $R5,=$R6
001812  07B5  0903                    BE      >$+2+$AF
001813  07B6  F380 11CB                LNJ     $B7,<ERROR          BVBI, BVBF, OR CCB LIST RESET ERROR
001814  07B8  8AD3                    INC     =$R3                BUMP TO NEXT CHANNEL
001815  07B9  0F61 FF3A                B       1ZTST6
001816                          *
001817  07BB  C005              DTPT    DC      2'C005'             RCV DATA PTR
001818  07BC  C025                    DC      2'C025'             XMIT DATA POINTER
001819  07BD  0000                    DC      0                   END OF LIST
001820                          *----------------------------------------------------------
001821                          *
001822                          *SUBROUTINES FOR INITIALIZE TEST
001823                          *
001824                          *
001825                          *     BLOCK READ A CHANNEL'S LCT AREA TO CHECK IT'S CLEAR
001826                          *
001827  07BE  C853              BRLCLR  LDR     $R4,=$R3
001828  07BF  4F20                      MLV     $R4,X'20'           FORM RAM ADDRESS OF LCT AREA
001829  07C0  CF00 07CB                 STR     $R4,<1ZTST4
001830  07C2  BF00 1347                 STR     $R3,<IPR
001831  07C4  3B00 07C7                 BEVN    $R3,<1ZTST2
001832  07C6  88D3                      DEC     =$R3
001833                          *
001834  07C7  9380 1009         1ZTST2  LNJ     $B1,<RDATA          READ RAM
001835  07C9  2800                      DC      <RTB                RECEIVE BUFFER
001836  07CA  0020                      DC      X'20'
001837  07CB  0000              1ZTST4  DC      0                   RAM ADDRESS
001838  07CC  0000                      DC      0                   START ON EVEN
001839                          *
001840  07CD  B800 1347                 LDR     $R3,<IPR
001841                          *
001842                          * CHECK ZERO'S WERE READ IN
001843                          *
001844  07CF  8756                      CL      =$R6
001845  07D0  2C02                      LDV     $R2,=2              START WITH WORD 2
001846  07D1  D820 2800         1ZTST1  LDR     $R5,<RTB,$R2
001847  07D3  D956                      CMR     $R5,=$R6
001848  07D4  0903                      BE      >$+2+$AF
001849  07D5  F380 11CB                 LNJ     $B7,<ERROR          LCT NOT CLEARED TO ZERO
001850  07D7  8AD2              1ZTST7  INC     =$R2
001851  07D8  AF57                      STR     $R2,=$R7
001852                          *
001853  07D9  3B83                      BODD    $R3,>1ZTST3
001854  07DA  2D04                      CMV     $R2,=X'4'           CHECK FOR READ COMMAND
001855  07DB  097C                      BE      >1ZTST7             OR CONTROL WORD
001856  07DC  2D10              1ZTST3  CMV     $R2,=X'10'
001857  07DD  09F4                      BNE     >1ZTST1
001858  07DE  8384                      JMP     $B4                 ALL DONE
001859                          *
001860                          *
001861                          * DO TWO INPUT NEXT STATUS TO GET THE INPUT
001862                          * NEXT STAT POINTER AWAY FROM IT'S INITIAL STATE
001863                          *
001864  07DF  C380 10DA         ADV2    LNJ     $B4,<INXT           INPUT NEXT STATUS
001865  07E1  C380 10DA                 LNJ     $B4,<INXT           INPUT NEXT STATUS
001866  07E3  8381                      JMP     $B1
001867
001868                          * READ RANGE TO INSURE INPUT NEXT STATUS POINTER IS RESET
001869                          *
001870  07E4  C800 13BF         CHKRNG  LDR     $R4,<CON14          INPUT NEXT STATUS
001871  07E6  C380 10EC                 LNJ     $B4,<CGSCH          MODIFY FOR CHANNEL
001872  07E8  8040 0B5F                 IO      TEMP,=$R4           INPUT NEXT TO GET TO FIRST CCB
        07EA  0054
001873  07EB  0703                      BIOT    >$+2+$AF
001874  07EC  F380 11CB                 LNJ     $B7,<ERROR
001875                          * NOW INPUT RANGE, SHOULD BE = 1 IF POINTER WAS RESET
001876  07EE  4EF2                      ADV     $R4,=-X'E'
001877  07EF  8055                      IO      =$R5,=$R4           INPUT RANGE
        07F0  0054
001878  07F1  0703                      BIOT    >$+2+$AF
001879  07F2  F380 11CB                 LNJ     $B7,<ERROR          COMMAND WAS NAKED
001880  07F4  6C01                      LDV     $R6,=1
001881  07F5  D956                      CMR     $R5,=$R6
001882  07F6  0903                      BE      >$+2+$AF
001883  07F7  F380 11CB                 LNJ     $B7,<ERROR          INPUT STAT POINTER NOT..
001884                          *                                   POINTING AT FIRST CCB
001885  07F9  8381                      JMP     $B1
001886                          *
001887                          * SET UP 4 CCB'S
001888                          *
001889  07FA  4C01              SET4    LDV     $R4,1
001890  07FB  C380 1102         SETA    LNJ     $B4,<MCCB           MAKE CCB
001891  07FD  2800                      DC      <RTB
001892  07FE  0040                      DC      X'40'               VALID CCB BIT
001893  07FF  8AD4                      INC     =$R4
001894  0800  4D05                      CMV     $R4,5
```

```
001895  0801  09FA                    BNE     >SETA
001896  0802  8381                    JMP     $B1
001897                          *
001898                          *
001899                          *-----------------------------------------------------------
001900                          *
001901                          *  STOP IO
001902                          *
001903  0803  9870 4B20          STIORC LDR     $R1,=A'K '
001904  0805  C380 13E4                 LNJ     $B4,<PLB                REPORT TEST
001905  0807  C380 106D                 LNJ     $B4,<GENITZ
001906  0809  8753                      CL      =$R3                    R3 TRACKS CHANNEL
001907  080A  C380 105D          RPSIO  LNJ     $B4,<FLN                FIND ACTIVE LINE
001908  080C  8382                      JMP     $B2                     NO MORE LINES TO TEST
001909                          * SET UP CONDITIONS FOR STOP IO TEST
001910  080D  C360 04BA                 LNJ     $B4,<LCTGRB             FILL  LCT AREA WITH DATA
001911  080F  B380 0F7D                 LNJ     $B3,<SETLCT             SEND OUT LCT BYTES
001912  0811  2400                      DC      <SDB
001913  0812  4C01                      LDV     $R4,=1                  RANGE
001914  0813  C380 1102                 LNJ     $B4,<MCCB               DUMMY CCB
001915  0815  2800                      DC      <RTB
001916  0816  0400                      DC      X'400'                  RAM ADDRESS
001917                          *
001918                          * GIVE STOP I/O
001919                          *
001920  0817  C360 10F0                 LNJ     $B4,<CHCT               DO CHANNEL CONTROL
001921  0819  0F82                      B       >$+2
001922  081A  2000                      DC      2'2000'                 STOP IO
001923                          *
001924  081B  C380 117D                 LNJ     $B4,<DLAY               DELAY
001925                          *
001926                          * CHECK THAT STOP I/O OPERATED CORRECTLY
001927                          *
001928  081D  C380 10DA                 LNJ     $B4,<INXT               INPUT NEXT STATUS
001929                          *
001930  081F  E870 1000                 LDR     $R6,=X'1000'            STAT COMPLETE BIT
001931  0821  8265                      LB      =$R5,=X'1000'
      0822  1000
001932  0823  0503                      BBT     >$+2+$AF
001933  0824  F380 11CB                 LNJ     $B7,<ERROR              STOP IO DIDN'T CAUSE STATUS COMPLETE
001934                          *
001935                          * INPUT LCT STATUS
001936                          *
001937  0826  C380 0836                 LNJ     $B4,<ILCTST
001938  0828  8756                      CL      =$R6
001939  0829  D956                      CMR     $R5,=$R6
001940  082A  0903                      BE      >$+2+$AF
001941  082B  F380 11CB                 LNJ     $B7,<ERROR              STOP IO DIDN'T CLEAR LCT STATUS
001942                          *
001943                          * DO A CHANNEL INITIALIZE IN PREPARATION FOR THE NEXT TEST
001944                          *
001945  082D  C360 10F0                 LNJ     $B4,<CHCT
001946  082F  0F82                      B       >$+2
001947  0830  0000                      DC      0                       CHANNEL INITIALIZE
001948  0831  C380 117D                 LNJ     $B4,<DLAY               TIME DELAY TO ALLOW INITIALIZE TO COMPLETE
001949                          *
001950  0833  8AB3                      INC     =$R3                    SET FOR NEXT CHANNEL NUMBER
001951  0834  0F81 FFD5                 B       RPSIO
001952                          *-----------------------------------------------------------
001953                          *
001954                          * SUBROUTINES FOR STOP IO
001955                          *
001956                          *
001957                          * INPUT LCT STATUS TO R5
001958                          *
001959  0836  8F40 0B3D          ILCTST SAVE    SAV2,=Z'AA88'           R4,B4
      0838  AA88
001960  0839  E870 1137                 LDR     $R6,=Z'1137'
001961  083B  A870 1037                 LDR     $R2,=Z'1037'
001962  083D  3600 0843                 BEVN    $R3,<ILC1               BRANCH IF RCV CHAN
001963  083F  AA70 2000                 ADD     $R2,=Z'2000'
001964  0841  EA70 2000                 ADD     $R6,=Z'2000'
001965  0843  9852              ILC1   LDR     $R1,=$R2                GET BYTE ADDRESS OF MSB OF STAT
001966  0844  C380 0DC3                 LNJ     $B4,<OUTLCT             OUTPUT BYTE
001967  0846  C380 1121                 LNJ     $B4,<INBYTE             INPUT BYTE TO R5
001968  0848  DF00 1347                 STR     $R5,<TPR
001969  084A  9856                      LDR     $R1,=$R6
001970  084B  C380 0DC3                 LNJ     $B4,<OUTLCT             OUTPUT BYTE
001971  084D  C380 1121                 LNJ     $B4,<INBYTE             INPUT LSB OF STAT
001972  084F  5048                      SUR     $R5,8                   MOVE OVER
001973  0850  D400 1347                 OR      $R5,<TPR                FORM WHOLE STAT WORD
001974  0852  8FC0 0B21                 RSTR    SAV2,=Z'AA88'
      0854  AA88
001975  0855  8384                      JMP     $B4                     EXIT SUBROUTINE
001976                          *-----------------------------------------------------------
001977                          *
001978                          *
001979                          *
001980                          * RECEIVE TESTS
001981                          *
001982                          *
001983                          *
001984                          *   1. EVEN BYTE START, EVEN RANGE, EOR TERMINATION
001985                          *
001986  0856  9870 4C31          RC0    LDR     $R1,=A'L1'
001987  0858  C380 13E4                 LNJ     $B4,<PLB
001988  085A  C380 08E6                 LNJ     $B4,<RCVTST             RECEIVE TEST
001989  085C  0F86                      B       >RC1                    END RETURN
001990  085D  0000                      DC      0                       EVEN BYTE BOUNDRY
001991  085E  0006                      DC      6                       CCB RANGE
001992  085F  0040                      DC      X'40'                   CCB CONTROL WORD
001993  0860  0A8E                      DC      <LCT2                   LCT TABLE
001994  0861  0006                      DC      6                       NUMBER OF BYTES TO XFER
001995                          *-----------------------------------------------------------
001996                          *
001997                          *   2. ODD START,<EVEN RANGE, EOR TERMINATION
001998                          *
001999  0862  9870 4C32          RC1    LDR     $R1,=A'L2'
002000  0864  C380 13E4                 LNJ     $B4,<PLB
002001  0866  C380 08E6                 LNJ     $B4,<RCVTST
002002  0868  0F86                      B       >RC2
002003  0869  0001                      DC      1                       ODD BYTE BOUNDRY
002004  086A  0006                      DC      6                       CCB RANGE
```

```
002005  086B  0040                    DC      X'40'                   CCB CONTROL
002006  086C  0A8E                    DC      <LCT2                   LCT TABLE
002007  086D  0006                    DC      6                       NUMBER OF BYTES TO XFER
002008                          *
002009                          *-----------------------------------------------------------------
002010                          *
002011                          *   3. EVEN START, EVEN RANGE, GNB TERMINATION
002012                          *
002013  086E  9870 4C33         RC2     LDR     $R1,=A'L3'
002014  0870  9780 13D5                 STM     $R1,<ERMG+1
002015  0872  C380 08E6                 LNJ     $B4,<RCVTST             RECEIVE TEST
002016  0874  0F86                      B       >RC3                    DONE
002017  0875  0000                      DC      0                       EVEN BYTE
002018  0876  0008                      DC      8                       CCB RANGE
002019  0877  0040                      DC      X'40'                   CCB CONTROL WORD
002020  0878  0A8D                      DC      <LCT2A                  LCT
002021  0879  0006                      DC      6                       NUMBER OF BYTES TO XFER
002022                          *-----------------------------------------------------------------
002023                          *
002024                          *   4. EVEN START, ODD RANGE, GNB TERMINATION
002025                          *
002026  087A  9870 4C34         RC3     LDR     $R1,=A'L4'
002027  087C  C380 13E4                 LNJ     $B4,<PLB
002028  087E  C380 08E6                 LNJ     $B4,<RCVTST             RECEIVE TEST
002029  0880  0F86                      B       >RC4                    DONE
002030  0881  0000                      DC      0                       EVEN BYTE
002031  0882  0005                      DC      5                       CCB RANGE
002032  0883  0040                      DC      X'40'                   CCB CONTROL WORD
002033  0884  0A8D                      DC      <LCT2A                  LCT
002034  0885  0003                      DC      3                       NUMBER OF BYTES TO XFER
002035                          *-----------------------------------------------------------------
002036                          *
002037                          * RECEIVE TESTS
002038                          *
002039                          *5. EVEN START, ODD RANGE, EOR TERMINATION
002040                          *
002041  0886  9870 4C35         RC4     LDR     $R1,=A'L5'
002042  0888  C380 13E4                 LNJ     $B4,<PLB
002043  088A  C380 08E6                 LNJ     $B4,<RCVTST             RECEIVE TEST
002044  088C  0F86                      B       >RC5                    RETURN
002045  088D  0000                      DC      0                       EVEN BYTE START
002046  088E  0005                      DC      5                       CCB RANGE
002047  088F  0040                      DC      X'40'                   CCB CONTROL
002048  0890  0A8E                      DC      <LCT2
002049  0891  0005                      DC      5                       NUMBER OF BYTES TO XFER
002050                          *-----------------------------------------------------------------
002051                          *
002052                          * CCB OVER RUN TEST
002053                          *
002054  0892  9870 4C36         RC5     LDR     $R1,=A'L6'
002055  0894  C380 13E4                 LNJ     $B4,<PLB
002056  0896  C380 08E6                 LNJ     $B4,<RCVTST
002057  0898  8382                      JMP     $B2                     EXIT TEST
002058  0899  0000                      DC      0
002059  089A  0003                      DC      3                       CCB RANGE
002060  089B  1040                      DC      X'1040'                 BIT 3 SET FOR FLAG FOR CCB SERVICE ERROR
002061  089C  0A8E                      DC      <LCT2
002062  089D  0005                      DC      5                       NUMBER OF BYTES  TO XFER
002063                          *
002064                          *
002065                          **
002066                          *
002067                          *-----------------------------------------------------------------
002068                          *
002069                          * TRANSMIT TESTS
002070                          *
002071                          *-----------------------------------------------------------------
002072                          *
002073                          *   1. EVEN START, EVEN RANGE, EOR TERM
002074                          *
002075  089E  9870 4D31         TRAN1   LDR     $R1,=A'M1'
002076  08A0  C380 13E4                 LNJ     $B4,<PLB
002077  08A2  C3C0 00DA                 LNJ     $B4,XTST                TRANSMIT TEST
002078  08A4  0F86                      B       >TRAN2                  RETURN
002079  08A5  0000                      DC      0                       EVEN BYTES
002080  08A6  0006                      DC      6                       CCB RANGE
002081  08A7  0060                      DC      =X'60'                  CCB CONTROL, LAST BLOCK
002082  08A8  0A87                      DC      <LCT1                   LCT
002083  08A9  0006                      DC      6                       NUMBER OF BYTES TO XFER
002084                          *
002085                          *-----------------------------------------------------------------
002086                          *
002087                          *   2. ODD START,<EVEN RANGE
002088                          *
002089  08AA  9870 4D32         TRAN2   LDR     $R1,=A'M2'
002090  08AC  C380 13E4                 LNJ     $B4,<PLB
002091  08AE  C3C0 00CE                 LNJ     $B4,XTST                TRANSMIT TEST
002092  08B0  0F86                      B       >TRAN3
002093  08B1  0001                      DC      1                       ODD BYTE
002094  08B2  0006                      DC      6                       CCB RANGE
002095  08B3  0060                      DC      =X'60'                  CCB CONTROL WORD, LAST BLOCK, VALID
002096  08B4  0A87                      DC      <LCT1                   LCT
002097  08B5  0006                      DC      6                       NUMBER OF BYTES TO XFER
002098                          *
002099                          *-----------------------------------------------------------------
002100                          *
002101                          *   3. EVEN START, EVEN RANGE, GNB TERMINATION
002102                          *
002103  08B6  9870 4D33         TRAN3   LDR     $R1,=A'M3'
002104  08B8  C380 13E4                 LNJ     $B4,<PLB
002105  08BA  C3C0 00C2                 LNJ     $B4,XTST                TRANSMIT TEST
002106  08BC  0F86                      B       >TRAN4                  RETURN
002107  08BD  0000                      DC      0                       EVEN BYTE
002108  08BE  0008                      DC      8                       CCB RANGE
002109  08BF  0060                      DC      X'0060'                 LAST BLOCK, VALID
002110  08C0  0A86                      DC      <LCT1A                  LCT
002111  08C1  0006                      DC      6                       NUMBER OF BYTES TO XFER
002112                          *
002113                          *-----------------------------------------------------------------
002114                          *
002115                          *   4. EVEN START, ODD RANGE, GNB TERMINATION
002116                          *
002117  08C2  9870 4D34         TRAN4   LDR     $R1,=A'M4'
```

```
002118  08C4  C380 13E4              LNJ   $B4,<PLB
002119  08C6  C3C0 00B6              LNJ   $B4,XTST          TRANSMIT TEST
002120  08C8  0F86                   B     >TRAN5            RETURN
002121  08C9  0000                   DC    0
002122  08CA  0008                   DC    8                 CCB RANGE
002123  08CB  0060                   DC    X'60'             CCB CONTROL, LAST BLOCK
002124  08CC  0A86                   DC    <LCT1A            LCT
002125  08CD  0005                   DC    5                 NUMBER OF BYTES TO XFER
002126                        *
002127                        *-----------------------------------------------------------
002128                        *
002129                        *   5. EVEN START, ODD RANGE, EOR TERMINATION
002130                        *
002131  08CE  9870 4D35       TRAN5 LDR   $R1,=A'M5'
002132  08D0  C380 13E4             LNJ   $B4,<PLB
002133  08D2  C3C0 00AA             LNJ   $B4,XTST          TRANSMIT TEST
002134  08D4  0F86                  B     >TRAN6
002135  08D5  0000                  DC    0                 EVEN START
002136  08D6  0005                  DC    5                 RANGE (CCB)
002137  08D7  0060                  DC    X'60'             CCB CONTROL , LAST  BLOCK
002138  08D8  0A87                  DC    <LCT1             LCT
002139  08D9  0005                  DC    5                 NUMBER OF BYTES TO XFER
002140                        *-----------------------------------------------------------
002141                        *
002142                        * CCB UNDER-RUN TEST
002143                        *
002144  08DA  9870 4D36       TRAN6 LDR   $R1,=A'M6'
002145  08DC  C380 13E4             LNJ   $B4,<PLB
002146  08DE  C3C0 009E             LNJ   $B4,XTST          TRANSMIT
002147  08E0  8382                  JMP   $B2               EXIT TEST
002148  08E1  0000                  DC    0                 EVEN  START
002149  08E2  0005                  DC    5                 CCB RANGE
002150  08E3  1040                  DC    X'1040'           NOT LAST CCB
002151  08E4  0A87                  DC    <LCT1             LCT
002152  08E5  0006                  DC    6                 XMIT 5
002153                        *
002154                        *-----------------------------------------------------------
002155                        *
002156                        * RECEIVE TEST
002157                        *
002158                        *   LNJ $B4,<RCVTST
002159                        *   B >NEXT
002160                        *   DC A              1= ODD BYTE
002161                        *   DC B              RANGE IN BYTES
002162                        *   DC C              CCB CONTROL WORD
002163                        *   DC D              POINTS TO LCT TABLE
002164                        *   DC E              NUMBER OF BYTES TO XFER
002165                        *
002166  08E6  8F40 0AAD       RCVTST SAVE   SAV4,=X'547C'     R1,3,4, B1,2,3,4,5
      08E8  547C
002167  08E9  9870 0200              LDR   $R1,=X'200'       P COUNTER START
002168  08EB  9F00 0A36              STR   $R1,<RCWR1
002169  08ED  9874                   LDR   $R1,+$B4          DUMMY TO INCREMENT B4
002170  08EE  9874                   LDR   $R1,+$B4          ODD/EVEN FLAG
002171  08EF  9F00 093B              STR   $R1,<RCV1
002172  08F1  9F00 0942              STR   $R1,<RCV2
002173  08F3  9F00 094B              STR   $R1,<RCV3
002174  08F5  9F00 0967              STR   $R1,<RCV4
002175                        *
002176  08F7  9874                   LDR   $R1,+$B4          PICK UP RANGE
002177  08F8  9F00 0947              STR   $R1,<RCV7
002178                        *
002179  08FA  9874                   LDR   $R1,+$B4          PICK UP CCB CONTROL WORD
002180  08FB  9F00 094C              STR   $R1,<RCV9
002181                        *
002182  08FD  9CF4                   LDB   $B1,+$B4          LCT POINTER
002183  08FE  9F80 094A              STB   $B1,<RCV10
002184                        *
002185  0900  9874                   LDR   $R1,+$B4          PICK UP NUMBER OF BYTES
002186  0901  9F00 0968              STR   $R1,<RCV8
002187  0903  9F00 093A              STR   $R1,<RCV5
002188  0905  9F00 0940              STR   $R1,<RCV6
002189  0907  8700 1346              CL    <FLAG
002190  0909  9900 0947              CMR   $R1,<RCV7         COMPARE RANGE WITHBYTES TO RCV
002191  090B  090C                   BE    >RCV13
002192  090C  0208                   BL    >RCV14
002193  090D  D800 0947              LDR   $R5,<RCV7
002194  090F  DF00 093A              STR   $R5,<RCV5
002195  0911  DF00 0968              STR   $R5,<RCV8
002196  0913  0F84                   B     >RCV13
002197  0914  4C20             RCV14 LDV   $R4,=X'20'        FLAG SET FOR NON ZERO RANGE RESIDUE
002198  0915  CF00 1346              STR   $R4,<FLAG
002199  0917  1006             RCV13 SOL   $R1,8             FORM RANGE
002200  0918  9670 003F              XOR   $R1,=X'3F'
002201  091A  9F00 0A8E              STR   $R1,<LCT2
002202                        *
002203  091C  8753                   CL    =$R3
002204  091D  C380 105D       NXRCV LNJ   $B4,<FLN          FIND ACTIVE LINE
002205  091F  0F82                   B     >$+2
002206  0920  0F83                   B     >$+2+$AF
002207  0921  0F81 0057              B     RCEND             NO MORE LINES
002208  0923  3B03                   BEVN  $R3,>RCV12        BRANCH MEANS RCV CHAN
002209  0924  8AD3                   INC   =$R3
002210  0925  0FF8                   B     >NXRCV            TRY NEXT
002211  0926  C853             RCV12 LDR   $R4,=$R3
002212  0927  4F20                   MLV   $R4,=X'20'        FORMS LCT START
002213  0928  4E39                   ADV   $R4,=57
002214  0929  CF00 0941              STR   $R4,<RC11         RAM ADDRESS WHERE DATA GOES
002215  092B  BF00 0939              STR   $R3,<RCDATA
002216                        *
002217                        * FILL RECEIVE BUFFER WITH DEFAULT
002218                        *
002219                                CALL  ZV$F,RTB,DFLT,BUFRNG
      092D  FBC0 0003
      092F  D380 0000         X
      0931  0F60
      0932  2800
      0933  133A
      0934  1340
002220  0935  C380 106D              LNJ   $B4,<GEN1I2       MLCC GENERAL INITIALIZE
002221                        * FILL SEND BUFFER WITH SB DATA FOR COMPARISON LATER
002222  0937  C380 0A94              LNJ   $B4,<FDTA         FILL ASCENDING DATA
002223  0939  0000             RCDATA DC    0               STARTING BYTE
```

```
002224  093A  0000            RCV5   DC     0                    RANGE IN BYTES
002225  093B  0000            RCV1   DC     0                    1 = START ON RIGHT BYTE
002226                        *
002227                        *
002228                        * BLOCK WRITE DATA TO RAM. DURING RECEIVE  THIS DATA WILL BE
002229                        * TRANSFERED TO CPU BY THE CHANNEL PROGRAM.
002230                        *
002231  093C  8AD3                   INC    =$R3                 GET XMIT HALF OF LINE
002232                        *
002233  093D  9380 0FDA               LNJ    $B1,<SDATA           SEND DATA
002234  093F  2400                   DC     <SDB                 FROM HERE
002235  0940  0000            RCV6   DC     0                    RANGE IN BYTES
002236  0941  0000            RC11   DC     0                    RAM ADDRESS
002237  0942  0000            RCV2   DC     0                    1 = START ON RIGHT BYTE
002238                        *
002239  0943  88D3                   DEC    =$R3                 GET BACK TO RECEIVE
002240                        *
002241                        * RECEIVE A BLOCK OF DATA
002242                        *
002243  0944  C380 0A12               LNJ    $B4,<RCV
002244  0946  2800                   DC     <RTB                 READ TO HERE
002245  0947  0000            RCV7   DC     0                    RANGE
002246  0948  12AC                   DC     <RCDTC1              CHANNEL PROGRAM ADDRESS
002247  0949  005E                   DC     (RCRT1-RCDTC1)*2     CHANNEL PROGRAM RANGE
002248  094A  1341            RCV10  DC     <DUMMY               LCT PARAMETER TABLE
002249  094B  0000            RCV3   DC     0                    1= START ON RIGHT BYTE
002250  094C  0000            RCV9   DC     0                    CCB CONTROL WORD
002251                        *
002252                        * READ AND  CHECK STATUS
002253                        *
002254  094D  C380 10DA               LNJ    $B4,<INXT            INPUT NEXT STATUS
002255  094F  E870 1000               LDR    $R6,=X'1000'         CCB COMPLETE BIT
002256  0951  E600 1346               XOR    $R6,<FLAG            NON ZERO RANGE
002257  0953  D956                   CMR    $R5,=$R6             RESIDUE BIT
002258  0954  0903                   BE     >$+2+$AF
002259  0955  F380 11CB               LNJ    $B7,<ERROR           STATUS ERROR
002260  0957  8756                   CL     =$R6
002261  0958  82C0 FFF3               LB     RCV9,=X'1000'        GET FLAG FOR CCB ERROR
        095A  1000
002262  095B  8A56                   LBS    =$R6,=X'800'         SET CCB SERVICE ERROR BIT
        095C  0800
002263                        *
002264                        * READ LCT STATUS
002265                        *
002266  095D  C380 0836               LNJ    $B4,<ILCTST          READ LCT STATUS
002267  095F  D570 0800               AND    $R5,=X'800'          STRIP TO SERVICE ERROR
002268  0961  D956                   CMR    $R5,=$R6
002269  0962  0903                   BE     >$+2+$AF
002270  0963  F380 11CB               LNJ    $B7,<ERROR           CCB SERVICE ERROR BIT  IN WRONG STATE
002271                        *
002272                        * CHECK DATA
002273                        *
002274  0965  D380 0A5B               LNJ    $B5,<CHKBW           CHECK ROUTINE
002275  0967  0000            RCV4   DC     0                    1 = START ON RIGHT BYTE
002276  0968  0000            RCV8   DC     0                    RANGE IN BYTES
002277                        *
002278                        * CHECK DATA BUFFER POINTER CLEARED TO 0
002279                        *
002280  0969  9870 0537               LDR    $R1,=X'0537'
002281  096B  C380 0DC3               LNJ    $B4,<OUTLCT          OUTPUT BYTE
002282  096D  C380 1121               LNJ    $B4,<INBYTE          INPUT BYTE
002283  096F  D570 C000               AND    $R5,=Z'C000'         STRIP TO DATA PTR
002284  0971  8756                   CL     =$R6
002285  0972  D956                   CMR    $R5,=$R6
002286  0973  0903                   BE     >RCV11
002287  0974  F380 11CB               LNJ    $B7,<ERROR           GNB DIDN'T RESET DATA PTR'S
002288  0976  8AD3            RCV11  INC    =$R3
002289                        *
002290  0977  0F81 FFA5               B      NXRCV                DO NEXT LINE
002291                        *
002292  0979  8FC0 0A1A        RCEND  RSTR   SAV4,=X'547C'
        097B  547C
002293  097C  8384                   JMP    $B4
002294                        *
002295                        *-------------------------------------------------------------
002296                        * DATA TRANSMIT TEST
002297                        *
002298                        *   LNJ $B4,XTST
002299                        *   B >NEXT          EXIT BRANCH
002300                        *   DC 0             1 = ODD BYTE
002301                        *   DC X             RANGE IN BYTES
002302                        *   DC Y             CCB CONTROL WORD
002303                        *   DC Z             POINTER TO LCT PARAMETER TABLE
002304                        *   DC W             NUMBER OF BYTES TO XFER
002305                        *
002306                        *
002307  097D  8F40 0A16        XTST   SAVE   SAV4,=X'547C'         R1,3,4, B1,2,3,4,5
        097F  547C
002308  0980  9874                   LDR    $R1,+$B4             DUMMY TO INCREMENT B4
002309  0981  9874                   LDR    $R1,+$B4             PICK UP EVEN, ODD FLAG
002310  0982  9F00 09C0               STR    $R1,<CHST1           STORE FLAG FOR ODD OR EVEN BYTE
002311  0984  9F00 09C8               STR    $R1,<CHST2
002312  0986  9F00 09F8               STR    $R1,<CHST3
002313  0988  9F00 09FC               STR    $R1,<CHST4
002314  098A  9874                   LDR    $R1,+$B4             PICK UP RANGE
002315  098B  9F00 09C4               STR    $R1,<CHRG2
002316                        *
002317  098D  8752                   CL     =$R2
002318  098E  9874                   LDR    $R1,+$B4             PICK UP CCB CONTROL WORD
002319  098F  82D1                   LB     =$R1,X'40'           GET LAST BLOCK BIT
        0990  0040
002320  0991  0503                   BBT    >XTST2               BRANCH IF NO UNDER-RUN
002321  0992  A870 0800               LDR    $R2,=X'800'          UNDER-RUN EXPECTED
002322  0994  9F00 09C9        XTST2  STR    $R1,<CCBSAV
002323  0996  AF00 1346               STR    $R2,<FLAG            0 MEANS NO CCB UNDER-RUN
002324                        *
002325  0998  9CF4                   LDB    $B1,+$B4             PICK UP LCT POINTER
002326  0999  9F80 09C7               STB    $B1,<BWLCT
002327                        *
002328  099B  9874                   LDR    $R1,+$B4             PICK UP NUMBER OF BYTES
002329  099C  9F00 09FD               STR    $R1,<CHRG4           NUMBER OF BYTES TO CHECK
002330  099E  9900 09C4               CMR    $R1,<CHRG2
002331  09A0  0383                   BLE    >XTST3
```

```
002332  09A1  9800 09C4            LDR   $R1,<CHRG2
002333  09A3  9F00 09F6  XTST3     STR   $R1,<CHRG3
002334  09A5  9F00 09BF            STR   $R1,<CHRG1
002335  09A7  9800 09FD            LDR   $R1,<CHRG4
002336  09A9  1008                SUL   $R1,8
002337  09AA  9670 003F            XOR   $R1,=X'3F'
002338  09AC  9F00 0A87            STR   $R1,<LCT1
002339                            *
002340  09AE  8753                CL    =$R3
002341  09AF  C380 105D  NXWRT    LNJ   $B4,<FLN              FIND ACTIVE LINE
002342  09B1  0F82                B     >$+2
002343  09B2  0F83                B     >$+2+$AF
002344  09B3  0F81 005A            B     BWEND                 NO MORE LINES
002345  09B5  3883                BODD  $R3,>XTST1            BRANCH MEANS XMIT CHAN
002346  09B6  8AD3                INC   =$R3
002347  09B7  0FF8                B     >NXWRT                TRY NEXT
002348  09B8  BF00 09BE  XTST1    STR   $R3,<EODATA
002349  09BA  C380 106D            LNJ   $B4,<GENITZ           GENERAL INITIALIZE
002350                            *
002351  09BC  C380 0A94            LNJ   $B4,<FDTA             FILL SDB WITH ASCENDING DTA
002352  09BE  0000        EODATA  DC    0                     FIRST DATA WORD
002353  09BF  0005        CHRG1   DC    5                     RANGE IN BYTES
002354  09C0  0000        CHST1   DC    0                     1= START ON RIGHT BYTE
002355                            *
002356  09C1  C380 0A50            LNJ   $B4,<WRT
002357  09C3  2400                DC    <SDB                  SEND BUFFER
002358  09C4  0005        CHRG2   DC    5                     RANGE
002359  09C5  127C                DC    <TRANM1               CHANNEL PROGRAM
002360  09C6  0060                DC    (RCDTC1-TRANM1)*2     RANGE OF XMIT PROGRAM
002361  09C7  1341        BWLCT   DC    <DUMMY                LCT PROTOTYPE
002362  09C8  0000        CHST2   DC    0                     0 = EVEN START
002363  09C9  0000        CCBSAV  DC    0                     CCB CONTROL WORD STORED HERE
002364                            *
002365                            * READ AND CHECK STATUS
002366                            *
002367  09CA  8756                CL    =$R6
002368  09CB  C380 10DA            LNJ   $B4,<INXT             INPUT NEXT STATUS
002369  09CD  82C0 FFFB            LB    CCBSAV,=X'0020'       GET LAST BLOCK BIT
        09CF  0020
002370  09D0  8A56                LBS   =$R6,=Z'0020'         SET CCB LAST BIT TO EXPECTED STATE
        09D1  0020
002371  09D2  8280 09C9            LB    <CCBSAV,=X'1000'
        09D4  1000
002372  09D5  0583                BBF   >NXT1                 BRANCH IF NOT UNDERUN TEST
002373  09D6  D570 1000            AND   $R5,=Z'1000'
002374  09D8  E470 1000  NXT1     OR    $R6,=Z'1000'          SET STAT  COMP. BIT
002375  09DA  D956                CMR   $R5,=$R6              IS VS SB
002376  09DB  0903                BE    >$+2+$AF
002377  09DC  F380 11CB            LNJ   $B7,<ERROR            STATUS ERR AFTER XMIT
002378  09DE  82C0 FFEA            LB    CCBSAV,=X'1000'       GET CCB UNDERRUN FLAG
        09E0  1000
002379  09E1  8756                CL    =$R6
002380  09E2  8A56                LBS   =$R6,=X'600'          CCB SERVICE ERROR BIT
        09E3  0800
002381                            ***
002382                            *
002383                            * READ LCT STATUS
002384                            *
002385                            ***
002386  09E4  C380 0836            LNJ   $B4,<ILCTST           INPUT LCT STATUS
002387  09E6  D570 0800            AND   $R5,=Z'0800'          STRIP TO CCB SERVICE ERROR
002388  09E8  D956                CMR   $R5,=$R6
002389  09E9  0903                BE    >$+2+$AF
002390  09EA  F380 11CB            LNJ   $B7,<ERROR            CCB SERVICE ERROR BIT IN WRONG STATE
002391                            *
002392                            * READ DATA BACK
002393                            *
002394  09EC  C853                LDR   $R4,=$R3
002395  09ED  88D4                DEC   =$R4                  GET RCV CHANNEL
002396  09EE  4F20                MLV   $R4,=X'20'
002397  09EF  4E39                ADV   $R4,=57               FORM START OF DATA IN RAM
002398  09F0  CF00 09F7            STR   $R4,<EOADD
002399  09F2  88D3                DEC   =$R3                  GET CHN NUMB FOR RCV HALF OF LINE
002400  09F3  9380 1009            LNJ   $B1,<RDATA            DO BLOCK READ
002401  09F5  2800                DC    <RTB                  INPUT BUFFER
002402  09F6  0005        CHRG3   DC    5                     RANGE
002403  09F7  0000        EOADD   DC    0                     RAM ADDRESS
002404  09F8  0000        CHST3   DC    0                     =1 IF BUFFER STARTS ON RIGHT BYTE
002405                            *
002406  09F9  8AD3                INC   =$R3                  GET XMIT CHAN BACK
002407                            *
002408                            * CHECK DATA
002409                            *
002410  09FA  D380 0A5B            LNJ   $B5,<CHKBW            CHECK ROUTINE
002411  09FC  0000        CHST4   DC    0                     0 = START LEFT BYTE
002412  09FD  0005        CHRG4   DC    5                     5 BYTES
002413                            *
002414                            * CHECK DATA POINTERS ARE RESET
002415                            *
002416  09FE  9870 2537            LDR   $R1,=X'2537'
002417  0A00  C380 0DC3            LNJ   $B4,<OUTLCT           OUT X'25' TO LOC X'37'
002418  0A02  C380 1121            LNJ   $B4,<INBYTE           INPUT BYTE
002419  0A04  D570 C000            AND   $R5,=Z'C000'          STRIP TO DATA PTR
002420  0A06  8756                CL    =$R6
002421  0A07  D956                CMR   $R5,=$R6
002422  0A08  0903                BE    >NXT2
002423  0A09  F380 11CB            LNJ   $B7,<ERROR            GNB DIDN'T RESET LR25, BITS 0,1, XMIT
002424  0A0B  8AD3        NXT2    INC   =$R3                  SET FOR NEXT CHANNEL
002425  0A0C  0F81 FFA2            B     NXWRT
002426                            *
002427  0A0E  8FC0 0985  BWEND    RSTR  SAV4,=X'547C'
        0A10  547C
002428  0A11  8384                JMP   $B4
002429                            *
002430                            *-----------------------------------------------------------
002431                            *         LNJ $B4,<RCV
002432                            *         DC  A     BUFFER ADDRESS
002433                            *         DC  B     RANGE VALUE
002434                            *         DC  C     CHANNEL PROGRAM ADDRESS
002435                            *         DC  D     RANGE OF CHANNEL PROGRAM
002436                            *         DC  LCT   LCT PARAMETER TABLE
002437                            *         DC  E     0,1 = EVEN, ODD BYTE START ADDRESS
002438                            *         DC  CCB   CCB CONTROL WORD
```

```
002439                                  *
002440                                  *
002441  0A12 8F40 0971          RCV     SAVE    SAV3,=Z'ECB4'           SAVE REGS.
        0A14 ECB4
002442  0A15 9870 0200                  LDR     $R1,=X'200'
002443  0A17 9F00 0A36                  STR     $R1,<RCWR1
002444  0A19 ACF4                       LDB     $B2,+$B4                LOAD $B2 WITH ADDRESS
002445  0A1A C874                       LDR     $R4,+$B4                LOAD $R4 WITH NUMBER OF BYTES
002446  0A1B D870 5555                  LDR     $R5,=Z'5555'            LOAD $R5 WITH INITIAL PATTERN
002447  0A1D DF02                       STR     $R5,$B2                 STORE AT TOP AND BOTTOM OF BUFFER
002448  0A1E CF52                       STR     $R4,=$R2
002449  0A1F 8AD2                       INC     =$R2                    FIND INDEX TO LAST WORD
002450  0A20 2041                       SUR     $R2,1                   R2 HAS WORDS + 1
002451  0A21 DF22                       STR     $R5,$B2.$R2
002452                                  *
002453  0A22 BCF4               WRT1    LDB     $B3,+$B4                LOAD $B3 WITH ADDRESS OF CHANNEL PROG.
002454  0A23 BF80 0A34                  STB     $B3,<RCV2M
002455  0A25 AF80 0A43                  STB     $B2,<RCV3M
002456  0A27 9874                       LDR     $R1,+$B4                RANGE OF CHANNEL PROGRAM
002457  0A28 9F00 0A35                  STR     $R1,<RCWR
002458  0A2A BCF4                       LDB     $B3,+$B4                PICK UP ADDRESS OF LCT TABLE
002459  0A2B BF80 0A3C                  STB     $B3,<RCLCT
002460  0A2D CFD5                       STB     $B4,=$B5
002461  0A2E BF00 1347                  STR     $R3,<IPR                FIND WRITE CHANNEL
002462  0A30 3BB2                       BUDD    $R3,>RCV1M
002463  0A31 8AD3                       INC     =$R3
002464  0A32 9380 0FDA          RCV1M   LNJ     $B1,<SDATA              WRITE OUT CHANNNEL PROGRAM
002465  0A34 1341               RCV2M   DC      <DUMMY                  CHAN PROG
002466  0A35 0000               RCWR    DC      0                       RANGE OF CHANNEL PROGRAM
002467  0A36 0200               RCWR1   DC      X'200'                  RAM ADDRESS
002468  0A37 0000                       DC      0                       RIGHT BYTE START
002469                                  *
002470  0A38 B800 1347                  LDR     $R3,<IPR
002471  0A3A B380 0F7D                  LNJ     $B3,<SETLCT             SEND OUT LCT BYTES
002472  0A3C 1341               RCLCT   DC      <DUMMY                  LCT TABLE ADDRESS
002473                                  *
002474  0A3D 9875                       LDR     $R1,+$B5                LOAD $R1 WITH INDEX
002475  0A3E A875                       LDR     $R2,+$B5                GET CCB CONTROL WORD
002476  0A3F AF00 0A44                  STR     $R2,<RCV4M
002477  0A41 C380 1102                  LNJ     $B4,<MCCB               FORM CCB FOR NORMAL OPERATION
002478  0A43 1341               RCV3M   DC      <DUMMY                  CPU DATA ADDRESS
002479  0A44 0010               RCV4M   DC      Z'0010'                 CCB CONTROL WORD
002480                                  *
002481  0A45 89D2                       CMZ     =$R2
002482  0A46 0803                       BAL     >DOLA                   BRANCH MEANS FLAG SET TO BYPASS START IO
002483  0A47 B380 063E                  LNJ     $B3,<STRT10
002484                                  *
002485  0A49 C380 117D          DOLA    LNJ     $B4,<DLAY               TIME DELAY
002486                                  *
002487  0A4B CCD5                       LDB     $B4,=$B5
002488  0A4C 8FC0 0937                  RSTR    SAV3,=Z'ECB4'
        0A4E ECB4
002489  0A4F 83B4                       JMP     $B4
002490                                  *
002491                                  *
002492                                  * ELNJE$B4,<WR1
002493                                  *EDCECPU ADDRESS OF BUFFER
002494                                  * EDCERANGE VALUE
002495                                  *EDCECPU ADRESS OF CHANNEL PROGRAM
002496                                  *           DC          RANGE OF CHANNEL PROGRAM
002497                                  *           DC          LCT  PARAMETER TABLE
002498                                  *EDCEO OR 1 ODD OR EVEN BYTE BUFFER ADDRESS START
002499                                  *           DC          CCB CONTROL WORD
002500                                  *
002501                                  *
002502  0A50 8F40 0933          WRT     SAVE    SAV3,=Z'ECB4'           SAVE REGS. R1,2,4,5, B2,3,5 1,<M
        0A52 ECB4
002503  0A53 ACF4                       LDB     $B2,+$B4                LOAD $B2 WITH CPU ADDRESS OF BUFFER
002504  0A54 C874                       LDR     $R4,+$B4                LOAD $R4 WITH NUMBER OF BYTES
002505  0A55 9870 0400                  LDR     $R1,=X'400'
002506  0A57 9F00 0A36                  STR     $R1,<RCWR1              STORE P COUNTER START
002507  0A59 0F81 FFC8                  B       WRT1
002508                                  *
002509                                  * CHECK SEND VS. RCV BUFFERS ON A BYTE BY BYTE BASIS.
002510                                  *
002511                                  *   LNJ $B5,<CHKBW
002512                                  *   DC 0                1= START ON RIGHT BYTE
002513                                  *   DC X                RANGE IN BYTES
002514                                  *
002515  0A5B 9875               CHKBW   LDR     $R1,+$B5                SET BYTE INDEX
002516  0A5C A875                       LDR     $R2,+$B5
002517  0A5D 8252                       NEG     =$R2                    RANGE
002518  0A5E BB80 2400                  LAB     $B3,<SDB                ADDRESS OF SEND BUFFER
002519  0A60 EB80 2800                  LAB     $B6,<RTB                ADDRESS OF RECEIVE BUFFER
002520                                  *
002521  0A62 D096               CHK1    LDH     $R5,$B6.$R1
002522  0A63 E0DF                       LDH     $R6,$B3.+$R1            GET SENT DATA
002523  0A64 D1D6                       CMH     $R5,=$R6
002524  0A65 0903                       BE      >$+2+$AF
002525  0A66 F380 11CB                  LNJ     $B7,<ERROR              DATA ERROR
002526  0A68 2780 0A62                  BINC    $R2,<CHK1
002527                                  *
002528                                  * DATA IS GOOD, CHECK 1 BYTE MORE TO SEE IF INPUT EXCEEDED RANGE
002529                                  *
002530  0A6A D096                       LDH     $R5,$B6.$R1
002531  0A6B 6C55                       LDV     $R6,=X'55'              WHAT SHOULD BE IN NEXT BYTE
002532  0A6C E955                       CMR     $R6,=$R5
002533  0A6D 0903                       BE      >$+2+$AF
002534  0A6E F380 11CB                  LNJ     $B7,<ERROR              INPUT EXCEEDED RANGE
002535                                  *
002536                                  * READ BACK P COUNTER VALUE
002537                                  *
002538  0A70 C380 0DDC                  LNJ     $B4,<RPVLU              READ P VALUE
002539                                  *
002540  0A72 B800 1347                  LDR     $R3,<IPR
002541  0A74 3B00 0A7F                  BEVN    $R3,<CHK2               BRANCH IF RECEIVE CHAN
002542  0A76 E870 0403                  LDR     $R6,=X'403'             SHOULD BE FOR TRANSMIT SIDE
002543  0A78 8280 09C9                  LB      <CCBSAV,=Z'1000'
        0A7A 1000
002544  0A7B 0586                       BBF     >CHK3
002545  0A7C E870 045A                  LDR     $R6,=X'45A'
002546  0A7E 0F83                       B       >CHK3
002547  0A7F E870 0203          CHK2    LDR     $R6,=X'203'             P VALUE FOR GOOD COMPLETION
```

```
002548  0A81  D956                CHK3   CMR    $R5,=$R6            COMPARE IS VS SB
002549  0A82  0903                       BE     >$+2+$AF
002550  0A83  F380 11CB                  LNJ    $B7,<ERROR          INCORRECT OPERATION OF CHAN PROGRAM
002551  0A85  8385                       JMP    $B5
002552                             *
002553                             * LCT TABLE OF XMIT TEST
002554                             *
002555  0A86  0138                LCT1A  DC     =X'0138'            BYTE 56 = 1 FOR GNB, =0 FOR EOR
002556  0A87  013F                LCT1   DC     =X'013F'            55 = COUNTER FOR BYTES
002557  0A88  0537                       DC     X'0537'
002558  0A89  0027                       DC     =X'27'              BYTE 39
002559  0A8A  0426                       DC     =X'0426'            BYTE 38
002560  0A8B  0125                       DC     X'0125'             PAUSE DISABLE
002561  0A8C  0000                       DC     0                   END OF TABLE
002562                             * LCT TABLE OF RCV TEST
002563  0A8D  0138                LCT2A  DC     =X'0138'            56 = FLG. 1 = GNB, = 0 FOR EOR
002564  0A8E  013F                LCT2   DC     =X'013F'            CONTER FOR BYTES
002565  0A8F  0537                       DC     X'0537'
002566  0A90  0007                       DC     =X'07'              P VALUE START = 2
002567  0A91  0206                       DC     =X'0206'
002568  0A92  0105                       DC     X'0105'             PAUSE DISABLE
002569  0A93  0000                       DC     0                   END OF LCT
002570                             *
002571                             * FILL SEND BUFFER WITH ASCENDING DATA + RECEIVE  BUFF WITH DEFAULT
002572                             * ONLY 10 LOC OF RECEIVE ARE FILLED WITH DEFAULT (5555).
002573                             *
002574                             * LNJ $B4,<FDATA
002575                             * DC DATA STARTING DATA
002576                             * DC RNG RANGE IN BYTES
002577                             * DC X    0 FOR LEFT BYTE START, 1 FOR RIGHT
002578                             *
002579  0A94  8F40 08EF            FDTA   SAVE   SAV3,=X'7440'        B1,R1,2,3,4
        0A96  7440
002580  0A97  9B80 2400                   LAB    $B1,<SDB            ADDRESS OF SEND DATA
002581  0A99  C870 5555                   LDR    $R4,=X'5555'        DEFAULT FOR FIRST AND LAST
002582  0A9B  CF00 2400                   STR    $R4,<SDB
002583  0A9D  A874                        LDR    $R2,+$B4            STARTING DATA
002584  0A9E  B874                        LDR    $R3,+$B4            RANGE
002585  0A9F  8253                        NEG    =$R3
002586  0AA0  9874                        LDR    $R1,+$B4
002587  0AA1  A7DD                FDTALP  STH    $R2,$B1,+$R1        STORE BYTE
002588  0AA2  8AD2                        INC    =$R2
002589  0AA3  3780 0AA1                   BINC   $R3,<FDTALP
002590  0AA5  C791                        STH    $R4,$B1,$R1         DEFAULT AT END
002591  0AA6  1CF6                        LDV    $R1,=-10
002592  0AA7  9B80 2800                   LAB    $B1,<RIB            ADDRESS OF RECEIVE
002593  0AA9  A870 5555                   LDR    $R2,=2'5555'        DEFAULT VALUE
002594  0AAB  AF71                FDTA1   STR    $R2,+$B1            STORE IN BUFFER
002595  0AAC  1780 0AAB                   BINC   $R1,<FDTA1
002596  0AAE  8FC0 08D5                   RSTR   SAV3,=X'7440'
        0AB0  7440
002597  0AB1  8384                        JMP    $B4
002598                             * ------------------------------------------------------------------
002599                             *
002600                             *
002601                             *       DATA INPUT AND CRC TEST
002602                             *
002603                             *
002604  0AB2  9870 4E20            CRC1   LDR    $R1,=A'N '
002605  0AB4  C380 13E4                   LNJ    $B4,<PLB            REPORT TEST
002606  0AB6  C380 106D                   LNJ    $B4,<GENIT2         MLCC GENERAL INITIALIZE
002607                             *
002608  0AB8  6C39                        LDV    $R6,=57             SET LOOP COUNT
002609  0AB9  EF00 1342                   STR    $R6,<COUNT
002610  0ABB  DB80 2400                   LAB    $B5,<SDB            ADDRESS OF START OF BUFFER
002611  0ABD  DF80 133F                   STB    $B5,<ADDS
002612  0ABF  A870 0100                   LDR    $R2,=X'100'
002613  0AC1  AF00 1353                   STR    $R2,<RANGE1
002614  0AC3  8753                NXCHC4 CL     =$R3
002615                             *
002616  0AC4  C380 105D            NXCHC  LNJ    $B4,<FLN            FIND LINE ON
002617  0AC6  0FFD                       B      >NXCHC4             NO MORE, GO BACK TO FIRST
002618  0AC7  3B03                       BEVN   $R3,>NXCHC1
002619  0AC8  8AD3                       INC    =$R3
002620  0AC9  0FFB                       B      >NXCHC
002621  0ACA  9800 1342            NXCHC1 LDR    $R1,<COUNT          GET LOOP COUNT
002622  0ACC  88D1                       DEC    =$R1
002623  0ACD  9F00 1342                   STR    $R1,<COUNT
002624  0ACF  89D1                       CMZ    =$R1
002625  0AD0  0302                       BG     >NXCHC2
002626  0AD1  8382                       JMP    $B2                 END OF THIS TEST
002627  0AD2  8700 135E            NXCHC2 CL     <MASK
002628  0AD4  8751                       CL     =$R1
002629  0AD5  8AD3                       INC    =$R3
002630                             *
002631                             *      PREPARE AND LOAD RAM INSTRUCTIONS
002632                             *
002633  0AD6  9380 0FDA                   LNJ    $B1,<SDATA
002634  0AD8  12DB                        DC     <RCRT1
002635  0AD9  0044                        DC     (RUD1-RCRT1)*2
002636  0ADA  0200                        DC     X'200'              RAM ADDRESS
002637  0ADB  0000                        DC     0                   EVEN BYTE START
002638                             *
002639                             *
002640  0ADC  88D3                        DEC    =$R3                GET RECEIVE CHANNEL
002641  0ADD  EB80 1320                   LAB    $B6,<CRC16
002642  0ADF  BB80 2400                   LAB    $B3,<SDB            ADDRESS OF CRC'S GENERATED
002643  0AE1  5C05                        LDV    $R5,=5              CHARACTER
002644  0AE2  E870 C002                   LDR    $R6,=2'C002'        SET CONF. IN CPT FOR CRC16
002645  0AE4  EF00 0B0F                   STR    $R6,<CPT3A
002646  0AE6  6C08                        LDV    $R6,=8              NUMBER OF BITS IN CHARACTER
002647  0AE7  F870 0100                   LDR    $R7,=X'100'          NUMBER OF CRC'S IN TABLE
002648  0AE9  9380 0B13                   LNJ    $B1,<TESC           GO DO TEST FOR CRC16
002649                             *
002650  0AEB  5C05                        LDV    $R5,=5              CHARACTER
002651  0AEC  E870 C202                   LDR    $R6,=2'C202'        SET CONF. IN CPT FOR CCITT
002652  0AEE  EF00 0B0F                   STR    $R6,<CPT3A
002653  0AF0  6C08                        LDV    $R6,=8              NUMBER OF BITS IN CHARACTER
002654  0AF1  9380 0B13                   LNJ    $B1,<TESC           GO DO TEST FOR CCITT
002655                             *
002656  0AF3  5C01                        LDV    $R5,=1
002657  0AF4  E870 00FF                   LDR    $R6,=X'FF'          CHANGE MASK
002658  0AF6  EF00 135E                   STR    $R6,<MASK
```

```
002659  0AF8  E870 C602              LDR   $R6,=Z'C602'        SET CONF. IN CPT FOR LRC8
002660  0AFA  EF00 0B0F              STR   $R6,<CPT3A
002661  0AFC  6C08                   LDV   $R6,=8
002662  0AFD  9380 0B13              LNJ   $B1,<TESC           GO DO TEST FOR LRC8
002663                          *
002664  0AFF  5C05                   LDV   $R5,=5
002665  0B00  8700 135E              CL    <MASK               RESET MASK
002666  0B02  E870 4402              LDR   $R6,=Z'4402'        SET CONF. IN CPT FOR CRC12
002667  0B04  EF00 0B0F              STR   $R6,<CPT3A
002668  0B06  6C06                   LDV   $R6,=6              NUMBER OF BITS IN CHARACTER
002669  0B07  9380 0B13              LNJ   $B1,<TESC           DO TEST FOR CRC12
002670                          *
002671  0B09  3E02                   ADV   $R3,=2              INCREMENT TO NEXT INPUT CHANNEL
002672  0B0A  0F80 0AC4              B     <NXCHC
002673                          *
002674                          * LCT FOR ABOVE TEST
002675                          *
002676  0B0C  0206              CPT3  DC    Z'0206'           P, MSB, RCV
002677  0B0D  0226                    DC    Z'0226'           P, XMIT , MSB
002678  0B0E  0105                    DC    Z'0105'           PAUSE DISABLE, RCV
002679  0B0F  C002              CPT3A DC    Z'C002'           CRC TYPE, RCV
002680  0B10  C022                    DC    Z'C022'           CRC, XMIT
002681  0B11  0125                    DC    Z'0125'           PAUSE DISABLE, XMIT
002682  0B12  0000                    DC    0                 END OF LCT
002683                          *
002684                          *
002685                          *
002686  0B13  C380 108C         TESC  LNJ   $B4,<CRC            DO SIMULATED CRC
002687                          *
002688  0B15  A870 0100               LDR   $R2,=X'100'
002689  0B17  DBA8 133F         TESCA LAB   $B5,*<ADDS.$R2      FORM ADDRESS FOR RECEIVE DATA
002690  0B19  DF80 133F               STB   $B5,<ADDS
002691  0B1B  9870 FF00               LDR   $R1,=-X'100'
002692  0B1D  BB98 0000         X     LAB   $B3,*<ZV$HR.$R1
002693  0B1F  0F00 0B13               NOP   <TESC
002694  0B21  DDD3                   CMB   $B5,=$B3
002695  0B22  0386                   BLE   >TESCC
002696  0B23  BB80 2400               LAB   $B3,<SDB            RESET ADDRESS
002697  0B25  BF80 133F               STB   $B3,<ADDS
002698  0B27  0FF0                   B     >TESCA              RESET MEM. POINTER, WE'RE OVER 64K.
002699  0B28  DF80 0B45         TESCC STB   $B5,<CP1
002700  0B2A  DF80 0B76               STB   $B5,<COMP+4+2*$AF
002701  0B2C  DF80 0B3C               STB   $B5,<CP2+4+$AF
002702  0B2E  A870 00F0               LDR   $R2,=X'F0'
002703  0B30  DBA8 133F               LAB   $B5,*<ADDS.$R2      ADDRESS FOR LAST CCB
002704  0B32  DF80 0B6C               STB   $B5,<CP3
002705  0B34  B380 0F7D               LNJ   $B3,<SETLCT         SEND LCT OUT FOR THIS LINE
002706  0B36  0B0C                   DC    <CP13
002707                          *
002708                          CP2   CALL  ZV$F,$,DFLT,RANGE1
        0B37  FBC0 0003
        0B39  D380 0000         X
        0B3B  0F80
        0B3C  0B37
        0B3D  133A
        0B3E  1353
002709                          *
002710                          * IN THIS TEST 4 CCB'S ARE INITIALLY SET UP AND ONE ADDITIONAL IS
002711                          * SET UP AFTER DATA TRANSFERS ARE STARTED. THE BASE OF THE CCB'S
002712                          * IS MOVED UP WITH EACH NEW BLOCK XFER SO THAT ALL AVAILABLE
002713                          * MEMORY WILL HAVE DATA TRANSFERED TO IT. THE CCB'S ARE AS FOLLOWS:
002714                          *
002715                          *        CCB  CPU ADD    RANGE (WORDS)
002716                          *
002717                          *         1     B          80
002718                          *         2     B+80       40
002719                          *         3     B+C0       20
002720                          *         4     B+E0       10
002721                          *         5     B+F0       10
002722                          *
002723                          * IN THE ABOVE B = @ADDS@. B STARTS AT 1900(HEX) AND
002724                          * INCREASES BY 100(HEX) AFTER EACH SET OF 5 CCB'S.
002725                          *
002726  0B3F  7CFC                    LDV   $R7,=-4
002727  0B40  C870 0100               LDR   $R4,=X'100'
002728  0B42  8751              TESB CL    =$R1
002729  0B43  C380 1102               LNJ   $B4,<MCCB           FORM CCB FOR CHANNEL RUN
002730  0B45  1341              CP1  DC    <DUMMY              CPU DATA ADDRESS
002731  0B46  0040                   DC    X'0040'           SET FOR NO RUPT
002732                          *
002733  0B47  4041                    SOR   $R4,1               SHIFT OVER RANGE
002734  0B48  9854                    LDR   $R1,=$R4
002735  0B49  BB98 0B45               LAB   $B3,*<CP1.$R1
002736  0B4B  BF80 0B45               STB   $B3,<CP1            SET FOR NEXT CCP
002737  0B4D  7780 0B42               BINC  $R7,<TESB
002738                          *
002739  0B4F  7CFB                    LDV   $R7,=-5
002740  0B50  8756                    CL    =$R6
002741                          *
002742  0B51  B380 063E         TESF  LNJ   $B3,<STRT10         START I/O
002743                          *
002744                          *
002745  0B53  C380 10DA         TESG  LNJ   $B4,<INXT           INPUT NEXT STATUS
002746  0B55  0F88                    B     >TESE
002747  0B56  C800 13C1         TESH  LDR   $R4,<CONT6
002748  0B58  C380 10EC               LNJ   $B4,<CGSCH
002749  0B5A  8755                    CL    =$R5
002750  0B5B  8055                    IO    =$R5,=$R4           INPUT STATUS
        0B5C  0054
002751                          *
002752                          *
002753  0B5D  82D5              TESE  LB    =$R5,=Z'1000'       SEE IF STATUS COMPLETE
        0B5E  1000
002754  0B5F  0506                    BBT   >TESD
002755  0B60  0F00 0B51               NOP   <TESF
002756  0B62  67F4                    BINC  $R6,>TESH           TRY AGAIN
002757                          *
002758  0B63  F380 11CB               LNJ   $B7,<ERROR          CCB NOT COMPLETE AFTER CHANNEL PROG. EXECUTIO
002759  0B65  7DFB              TESD  CMV   $R7,=-5
002760  0B66  0988                    BNE   >TESJ
002761  0B67  C870 0020               LDR   $R4,=X'20'          RANGE
002762  0B69  8751                    CL    =$R1                CLEAR INDEX
002763  0B6A  C380 1102               LNJ   $B4,<MCCB           MAKE CCB
```

```
002764  0B6C  1341                CP3     DC      <DUMMY                  FILLED IN BY PROGRAM
002765  0B6D  0060                        DC      X'0060'                 LAST BLOCK, VALID
002766  0B6E  7780 0B53            TESJ    BINC    $R7,<TESG               TEST FOR LAST
002767                             *
002768                             COMP    CALL    ZV$C,SDB,$,MASK,RANGE1,ERAR
        0B70  FBC0 0003
        0B72  D360 0000      X
        0B74  0F80
        0B75  2400
        0B76  0B70
        0B77  135E
        0B78  1353
        0B79  1360
002769  0B7A  89C0 07E5                    CMZ     ERAR
002770  0B7C  090F                        BE      >NOERC                  IF ERROR HALT WITH:
002771  0B7D  DC80 0B76                    LDB     $B5,<COMP+4+2*$AF
002772  0B7F  D800 1362                    LDR     $R5,<ERAR+2             '15' IN $R5
002773  0B81  E800 1361                    LDR     $R6,<ERAR+1             'SHOULD BE' IN $R6
002774  0B83  F800 1360                    LDR     $R7,<ERAR               WITH THE LOCATION OF ERROR IN BLOCK
002775  0B85  F380 11CB                    LNJ     $B7,<ERROR              ERROR IN CRC GENERATION, $B6 POINTS TO TYPE
002776                             * OF CRC+1. $B5 POINTS TO START OF RECEIVE BUFFER
002777                                     CALL    ZV$CO                   CONTINUE COMPARISON
        0B87  FBF0 0001
        0B89  D360 0000      X
002778  0B8B  C380 10F0            NOERC   LNJ     $B4,<CHCT               CHANNEL CONTROL
002779  0B8D  0F82                        B       >$+2
002780  0B8E  0000                        DC      0                       CHANNEL INIT.
002781                             *
002782  0B8F  F870 0100                    LDR     $R7,=X'100'
002783  0B91  BB80 2400                    LAB     $B3,<SDB
002784  0B93  8381                        JMP     $B1                     JUMP TO DO NEXT CRC TYPE
002785                             *-----------------------------------------------------------
002786                             *
002787                             *
002788                             *
002789                             *           DATA OUTPUT TEST
002790                             *
002791                             *
002792  0B94  9870 5020            OTPT    LDR     $R1,=A'P '
002793  0B96  C380 13E4                    LNJ     $B4,<PLB                REPORT TEST
002794  0B98  8753                        CL      =$R3                    INITIALIZE CHANNEL
002795  0B99  8751                        CL      =$R1
002796  0B9A  C380 106D                    LNJ     $B4,<GENIIZ             GENERAL INITIALIZE
002797                             *
002798  0B9C  C380 105D            OTPT2   LNJ     $B4,<FLN                FIND LINE ON
002799  0B9E  8382                        JMP     $B2                     NO SUBCH ON NO TEST
002800  0B9F  3B83                        BODD    $R3,>OTPT4              GET WRITE SUBCH. ONLY
002801  0BA0  8AD3                        INC     =$R3
002802  0BA1  0FFB                        B       >OTPT2
002803  0BA2  8700 135E            OTPT4   CL      <MASK
002804                             *
002805                             *     LOAD AND PACK RAM INSTRUCTIONS
002806                             *
002807  0BA4  9380 0FDA                    LNJ     $B1,<SDATA
002808  0BA6  12FB                        DC      <ROD1
002809  0BA7  0024                        DC      (VALTST-ROD1)*2         NUMBER OF RAM INSTRUCTIONS
002810  0BA8  0200                        DC      X'200'                  RAM STARTING ADDRESS
002811  0BA9  0000                        DC      0                       EVEN START ADDRESS
002812                             *
002813                             *
002814                             *
002815                             *           FORM RANDOM DATA TABLE TO BE OUTPUT
002816                             *
002817  0BAA  BB80 2400                    LAB     $B3,<SDB                LOAD $B3 WITH ADDRESS OF SDB
002818  0BAC  EB80 1320                    LAB     $B6,<CRC16              LOAD $B6 WITH ADDRESS OF CRC16
002819  0BAE  5C0A                        LDV     $R5,10                  INITIAL CHARACTER
002820  0BAF  6C0B                        LDV     $R6,=11                 NUMBER OF BITS IN CHAR.
002821  0BB0  F870 01F4                    LDR     $R7,=500                THE NUMBER OF WORDS MADE IN TABLE
002822  0BB2  C360 108C                    LNJ     $B4,<CRC
002823                             *
002824  0BB4  B380 0F7D            OTPT5   LNJ     $B3,<SETLCT             SEND LCT BYTES OUT
002825  0BB6  0B0C                        DC      <CPT3
002826                             *
002827  0BB7  C870 01F4                    LDR     $R4,=500                RANGE OF BYTES TO BE LOADED FOR CRC
002828  0BB9  C380 1102                    LNJ     $B4,<MCCB               FORM CCB FOR CHANNEL RUN
002829  0BBB  2400                        DC      <SDB                    XMIT BUFFER ADDRESS
002830  0BBC  0040                        DC      X'0040'                 NO RUPT
002831                             *
002832  0BBD  C870 00C8                    LDR     $R4,=200
002833  0BBF  C380 1102                    LNJ     $B4,<MCCB               MAKE CCB
002834  0BC1  24FA                        DC      <SDB+250
002835  0BC2  0040                        DC      X'40'
002836                             *
002837  0BC3  C870 0064                    LDR     $R4,=100
002838  0BC5  C380 1102                    LNJ     $B4,<MCCB               MAKE CCB
002839  0BC7  255E                        DC      <SDB+350                CPU ADDRESS
002840  0BC8  0040                        DC      X'40'                   VALID BIT ONLY
002841                             *
002842  0BC9  C870 0064                    LDR     $R4,=100
002843  0BCB  C380 1102                    LNJ     $B4,<MCCB               MAKE CCB
002844  0BCD  2590                        DC      <SDB+400                CPU ADDRESS
002845  0BCE  0040                        DC      X'40'                   VALID
002846                             *
002847                             *
002848  0BCF  B380 063E                    LNJ     $B3,<STRTIO             START I/O
002849                             *
002850                             * WAIT FOR FIRST CCB TO GET READY AND THEN GIVE 5TH CCB
002851                             *
002852  0BD1  C380 1141                    LNJ     $B4,<TEST               WAIT FOR STATUS COMPLETE
002853  0BD3  0F82                        B       >$+2                    RETURN
002854  0BD4  003C                        DC      60                      1/2 SEC TIMEOUT
002855                             *
002856  0BD5  82D5                        LB      =$R5,=Z'1000'           GET STAT COMPLETE BIT
        0BD6  1000
002857  0BD7  0500 0BDB                    BBT     <OTPT7                  IF TRUE GO SET UP CCB
002858  0BD9  F380 11CB                    LNJ     $B7,<ERROR              NO CCB COMPLETE ON TRANSMIT
002859                             *
002860  0BDB  C870 0064            OTPT7   LDR     $R4,=100                RANGE
002861  0BDD  C380 1102                    LNJ     $B4,<MCCB               MAKE CCB
002862  0BDF  25C2                        DC      <SDB+450                CPU ADDRESS
002863  0BE0  0060                        DC      X'60'                   VALID,LAST BLOCK
002864                             *
002865  0BE1  C380 117D                    LNJ     $B4,<DLAY               TIME DELAY
```

```
002866                          *
002867  0BE3  E853             *       LDR     $R6,=$R3                CALCULATE LCT CRC RESIDUE ADDRESS
002868  0BE4  6F20             *       MLV     $R6,=X'20'
002869  0BE5  6E03             *       ADV     $R6,=3
002870  0BE6  EF40 0010        *       STR     $R6,OTPT3
002871                          *
002872                          *        ACCUMULATE CRC OF RANDOM TABLE FORMED EARLIER
002873                          *
002874  0BE8  5CFF             *       LDV     $R5,=-1
002875  0BE9  BB80 2400        *       LAB     $B3,<SDB
002876  0BEB  EB80 1320        *       LAB     $B6,<CRC16
002877  0BED  6C08             *       LDV     $R6,8                   8 BIT CHARACTER
002878  0BEE  F870 01F4        *       LDR     $R7,=500                NUMBER IN TABLE
002879  0BF0  C380 108C        *       LNJ     $B4,<CRC
002880                          *
002881  0BF2  88D3             *       DEC     =$R3                    GET READ CHANNEL
002882  0BF3  9380 1009        *       LNJ     $B1,<RDATA              DO BLOCK READ
002883  0BF5  1341             *       DC      <DUMMY                  TO HERE
002884  0BF6  0002             *       DC      2                       RANGE =2
002885  0BF7  0000             OTPT3   DC      0                       RAM ADDRESS GOS HERE
002886  0BF8  0000             *       DC      0                       EVEN BYTE BOUNDRY
002887                          *
002888                          *
002889  0BF9  D800 1341        *       LDR     $R5,<DUMMY
002890  0BFB  5018             *       SCL     $R5,8
002891  0BFC  D900 1350        *       CMR     $R5,<CRCAC              ARE CRCS' EQUAL
002892  0BFE  0905             *       BE      >PRFR1                  YES DO NEXT CHANNEL
002893  0BFF  E800 1350        *       LDR     $R6,<CRCAC              'IS' IN $R5 'SB' IN $R6
002894                          *
002895  0C01  F380 11CB        *       LNJ     $B7,<ERROR              OUTPUT ERROR
002896                          *
002897  0C03  3E03             PRFR1   ADV     $R3,=3                  INCREMENT TO NEXT CHANNEL
002898  0C04  C380 105D        OTPT1   LNJ     $B4,<FLN
002899  0C06  8382             *       JMP     $B2
002900  0C07  3B80 0BB4        *       BODD    $R3,<OTPT5
002901  0C09  8AD3             *       INC     =$R3
002902  0C0A  0FFA             *       B       >OTPT1
002903                          *----------------------------------------------------------------------
002904                          *
002905                          *
002906                          * INTERRUPT DESCRIPTION
002907                          *
002908                          ****
002909                          * BEFORE RUNNING UNDER INTERRUPT MODE THE FOLLOWING
002910                          * TABLES MUST BE SET UP. ALL TABLE CONTAIN 16 ENTITIES
002911                          * CORRESPONDING TO THE 16 CHANNELS IN SEQUENCE. AN ENTITY
002912                          * IS EITHER A DATA OR ADDRESS CONSTANT.
002913                          *
002914                          *
002915                          *       INMB - SET UP TO CONTAIN THE NEG. NUMBER OF
002916                          *              INTERRUPTS EXPECTED PER CHANNEL.
002917                          *              IF IT IS SET TO A POS. NUMBER IT WILL
002918                          *              BE ASSUMED THAT NO INTERRUPTS ARE EXPECTED
002919                          *              ON THAT CHANNEL.
002920                          *
002921                          *       ILV - SET UP TO CONTAIN THE LEVEL ASSIGNED
002922                          *             TO EACH CHANNEL.
002923                          *
002924                          *       IST - SET UP TO CONTAIN THE STATUS EXPECTED
002925                          *             ON EACH CHANNEL AFTER INTERRUPTS.
002926                          *
002927                          *
002928                          * AFTER AN INTERRUPT TEST HAS BEEN TERMINATED (NORMALLY OR OTHERWISE)
002929                          * THERE IS ONE OTHER TABLE CONTAINING USEFUL INFORMATION
002930                          *
002931                          *       PRIST  - CONTAINS ORDER IN WHICH FIRST 16
002932                          *                INTERRUPTS OCCURED.
002933                          *
002934                          * THE PROGRAM RUNS AT 5 LOGICALLY DISTINCT LEVELS:
002935                          *
002936                          *       LEVEL 0 - USED FOR SETUPS AND DATA CHECKING.
002937                          *                 IT IS EXITED BY SCHEDULING LEVEL 63
002938                          *                 BY MEANS OF A "LEV" INSTRUCTION
002939                          *
002940                          *       LEVEL 3 - REAL TIME CLOCK TIMEOUT. MAJOR TESTS
002941                          *                 WITH INTERRUPTS TERMINATE BY A RTC
002942                          *                 TIMEOUT. THIS LEVEL EXITS BY SCHEDULING
002943                          *                 LEVEL 0.
002944                          *
002945                          *
002946                          *       LEVEL 4 -  CCB DISPATCHER AND INTERRUPT MONITOR.
002947                          *                  THIS ROUTINE DOES STATUS CHECKING AND
002948                          *                  DISPATCHES A NEW CCB FOR A CHANNEL
002949                          *                  IF NEEDED. LEVEL 4 EXITS TO LEV 63 BY
002950                          *                  A "LEV" INSTRUCTION.
002951                          *
002952                          *       LEVELS 5-62   -  16 OF THESE LEVELS (VARIABLE) ARE USED
002953                          *                        FOR THE 16 CHANNELS. EACH LEVEL
002954                          *                        POINTS TO THE SAME RE-ENTRANT
002955                          *                        CHANNEL INTERRUPT HANDLER.
002956                          *
002957                          *       LEVEL 63 -  A "DO NOTHING-WAIT" ROUTINE
002958                          *                   RESIDES AT THIS LEVEL.
002959                          *
002960                          * TO USE THE PROGRAM INTERRUPT STRUCTURE THE FOLLOWING
002961                          * STEPS MUST BE TAKEN
002962                          *
002963                          *       1. USE ROUTINE INTITZ. THIS INITIALIZES ALL
002964                          *       TABLES.
002965                          *
002966                          *       2. FILL TABLES WITH INFORMATION.
002967                          *
002968                          *       3. SETUP INTERRUPT VECTORS. (INITITZ PUTS
002969                          *       VECTORS FOR ERROR HANDLER IN LEVELS 5-62. IT
002970                          *       SETS UP VECTORS FOR LEV 0,3,4, AND 5 TO THIER
002971                          *       CORRECT VALUE.)
002972                          *       4. USE ROUTINE "RTC" TO SET UP RTC TO
002973                          *            DESIRED TIMEOUT VALUE AND TO START RTC.
002974                          *
002975                          *       5. TURN ON DESIRED CHANNELS.
002976                          *
002977                          *       6. SCHEDULE AND DISPATCH LEVEL 63.
002978                          *
```

```
002979                          * WHEN THE RTC TIMEOUT IS COMPLETE THE PROGRAM
002980                          * WILL RESUME AT THE INSTRUCTION FOLLOWING THE "LEV"
002981                          * WHICH DISPATCHED LEVEL 63.
002982                          *
002983                          *
002984                          * *****************
002985                          *
002986                          * FIRST INTERRUPT TEST
002987                          *
002988  0C0B  9870 5220         INST1   LDR     $R1,=A'R '
002989  0C0D  C380 13E4                 LNJ     $B4,<PLB                REPORT TEST
002990  0C0F  8753                      CL      =$R3
002991  0C10  8E70 0080                 LEV     =Z'0080'                SWITCH TO LEVEL 0
002992  0C12  C380 106D         INLOOP  LNJ     $B4,<GENITZ             GENERAL INITIALIZE TO MLCC
002993  0C14  C380 105D                 LNJ     $B4,<FLN                FIND ACTIVE LINE
002994  0C16  8382                      JMP     $B2                     NO MORE LEFT
002995  0C17  9380 0EC9                 LNJ     $B1,<INTITZ             INITIALIZE INTERRUPTS
002996  0C19  C380 0DCF                 LNJ     $B4,<CHINTZ             INITIALIZE CHANNEL
002997  0C1B  C380 117D                 LNJ     $B4,<DLAY               DELAY 25MS
002998                          *
002999  0C1D  B380 0F7D                 LNJ     $B3,<SETLCT             OUTPUT LCT
003000  0C1F  0CD1                      DC      <INILCT                 FROM HERE
003001                          *
003002                          * SET UP EXPECTED STATUS TABLE
003003  0C20  9870 5020                 LDR     $R1,=Z'5020'            STATUS COMPLETE
003004  0C22  9F30 26E6                 STR     $R1,<IST.$R3
003005                          *
003006  0C24  1C06                      LDV     $R1,=6                  PUT IN LEVEL OF 6 FOR CHANNEL
003007  0C25  9F30 26F6                 STR     $R1,<ILV.$R3
003008                          * SEND OUT CHANNEL PROGRAM
003009  0C27  BF00 1347                 STR     $R3,<IPR
003010  0C29  8953                      LBI     =$R3,=X'1'
        0C2A  0001
003011  0C2B  9380 0FDA                 LNJ     $B1,<SDATA              SEND PROGRAM
003012  0C2D  0CA7                      DC      <GNDPRG
003013  0C2E  0014                      DC      20                      20 BYTE
003014  0C2F  0200                      DC      =X'200'                 RAM ADDRESS
003015  0C30  0000                      DC      0                       STARTS ON EVEN BYTE IN CPU
003016  0C31  B800 1347                 LDR     $R3,<IPR
003017                          * SEND OUT CCB. IN THIS FIRST TEST THE "1" BIT (BIT 0) IS
003018                          * SET TO A "0". NO INTERRUPT SHOULD OCCUR FOR THIS FIRST TEST
003019  0C33  C870 0004                 LDR     $R4,=4                  ANY RANGE
003020  0C35  8751                      CL      =$R1
003021                          *
003022  0C36  C380 1102                 LNJ     $B4,<MCCB               SEND CCB
003023  0C38  2800                      DC      <RTB                    ANY ADDRESS
003024  0C39  0060                      DC      =X'0060'                LAST BLOCK BIT ONLY
003025                          *
003026                          * SEND OUT INTERRUPT LEVEL 6 + RETURN CHANNEL NUMBER TO LCT
003027                          *
003028  0C3A  C380 0DAD                 LNJ     $B4,<LEVRCH             OUTPUT LEVEL + RETURN CHANNEL
003029                          *
003030                          * SET UP INTERRUPT VECTOR IN CP
003031                          *
003032  0C3C  C380 104C                 LNJ     $B4,<SETVEC             SETS UP VECTOR FOR LEVEL
003033                          *                                       SET UP IN TABLE ILV
003034  0C3E  C800 13BD                 LDR     $R4,<CONT2              GET INPUT RANGE FC
003035  0C40  C380 10EC                 LNJ     $B4,<CGSCH              PUT IN ADDRESS
003036  0C42  CF00 0C52                 STR     $R4,<RNFC
003037                          *
003038                          * TURN ON CHANNEL
003039                          *
003040  0C44  9B80 0C53                 LAB     $B1,<LV63TA             GET ADDRESS TO CONTINUE
003041  0C46  9F80 0E9D                 STB     $B1,<L63SV2             PUT IN LEVEL 63 SAVE AREA
003042  0C48  C380 11BB                 LNJ     $B4,<RTC                SETUP RTC TO INTERRUPT
003043  0C4A  0006                      DC      6                       AFTER 50 MSEC
003044  0C4B  8F00 0E9F                 SAVE    <L63SV3,=X'7884'
        0C4D  7884
003045  0C4E  8E70 803F                 LEV     =Z'803F'                SUSPEND CURRENT AND GO TO 63
003046  0C50  0F80 0C5E                 B       <BKHK
003047  0C52  0000             RNFC    DC      0                       RANGE CONTROL WORD
003048                          *
003049  0C53  B380 1021         LV63TA  LNJ     $B3,<TRNON              TURN ON CHANNEL
003050  0C55  4000                      DC      =X'4000'                CHANNEL CONTROL WORD - START IO
003051  0C56  0000                      DC      0                       WORKS FOR XMIT + RECEIVE
003052  0C57  8E70 00BF         IDLP    LEV     =Z'00BF'                DUMM LEV TO GENERATE RINT
003053  0C59  8055                      IO      =$R5,<RNFC              INPUT RANGE
        0C5A  0000 0C52
003054  0C5C  0F80 0C57                 B       <IDLP
003055                          *
003056                          * PROGRAM RETURNS TO LEVEL 0 HERE IF NO ERRORS
003057  0C5E  0005              BKHK    RTCF                            TURN OFF REAL TIME CLOCK
003058                          *
003059                          * CHECK STATUS IS COMPLETE
003060                          *
003061  0C5F  C380 10DA                 LNJ     $B4,<INXT               INPUT NEXT STATUS TO R5
003062  0C61  82D5                      LB      =$R5,=Z'1000'
        0C62  1000
003063  0C63  0503                      BBT     >$+2+$AF
003064  0C64  F380 11CB                 LNJ     $B7,<ERROR              STATUS NOT COMPLETE AFTER
003065                          *                                       "GNB" ISSUED
003066                          *
003067                          * SET UP FOR NEXT TEST. SET UP CCB WITH 1 BIT = 1 (PERMITTING
003068                          * INTERRUPTS)
003069                          *
003070  0C66  8700 1346                 CL      <FLAG                   FLAG CLEARED FOR FIRST TIME THROUGH
003071  0C68  9380 0EC9         INST2   LNJ     $B1,<INTITZ             INITIALIZE PROGRAM FOR INTERRUPTS
003072  0C6A  9870 9020                 LDR     $R1,=Z'9020'
003073  0C6C  8980 1346                 CMZ     <FLAG
003074  0C6E  0983                      BNE     >INST4                  BRANCH IF SECOND TIME THRU
003075                          *
003076  0C6F  9870 5020                 LDR     $R1,=Z'5020'            STATUS COMPLETE
003077  0C71  3B03              INST4   BEVN    $R3,>NEXT4              BRANCH IF RCV CHANNEL
003078  0C72  9670 0020                 XOR     $R1,=Z'0020'
003079  0C74  9F30 26E6         NEXT4   STR     $R1,<IST.$R3            SET EXPECTED STATUS
003080                          *
003081  0C76  1C06                      LDV     $R1,=6                  CHANNEL TO HAVE LEVEL 6
003082  0C77  9F30 26F6                 STR     $R1,<ILV.$R3
003083                          *
003084  0C79  1CFF                      LDV     $R1,=-1
003085  0C7A  9F30 26D6                 STR     $R1,<INMB.$R3           SET TO EXPECT ONE INTERRUPT
003086  0C7C  C380 104C                 LNJ     $B4,<SETVEC             SET UP RUPT VECTOR IN CP
003087                          *
```

```
003088                          * SEND OUT CCB WITH 1 BIT SET
003089                          * THIS CCB NOT USED SECOND TIME THRU LOOP
003090                          *
003091  0C7E  4CFF                     LDV     $R4,=-1                 ANY RANGE
003092  0C7F  8751                     CL      =$R1                    CLEAR BYTE INDEX
003093  0C80  C380 1102         *      LNJ     $B4,<MCCB               MAKE CCB
003094  0C82  2800                     DC      <RTB                    ANY ADDRESS
003095  0C83  00E0                     DC      =Z'00E0'                INTERRUPT,<LAST BLOCK
003096  0C84  9B80 0C53                 LAB     $B1,<LV63TA             GET ADDRESS TO CONTINUE
003097  0C86  9F80 0E9D                 STB     $B1,<L63SV2             PUT IN LEVEL 63 SAVE AREA
003098  0C88  C380 11BB                 LNJ     $B4,<RTC                SET UP RTC TO INTERRUPT
003099  0C8A  0006                     DC      6                       AFTER 50 MSEC
003100  0C8B  8F00 0E9F                 SAVE    <L63SV3,=X'7884'
        0C8D  7884
003101  0C8E  8E70 803F                 LEV     =Z'803F'                SUSPEND CURRENT AND GO TO 63
003102                          *
003103                          * RETURN HERE AFTER INTERRUPTS HAVE OCCURED
003104                          *
003105  0C90  0005                     RTCF                            TURN OFF REAL TIME CLOCK
003106                          *                                       "GNB" ISSUED
003107                          * CHECK THAT INTERRUPT DID OCCUR
003108                          *
003109  0C91  9380 0F6A                 LNJ     $B1,<CHLV               CHECK 1 INTERRUPT OCCURED
003110  0C93  9800 1346                 LDR     $R1,<FLAG
003111  0C95  198F                     BNEZ    $R1,>INST3              BRANCH IF SECOND TIME THRU
003112  0C96  8A80 1346                 INC     <FLAG                   SET FIRST TIME THRU FLAG
003113  0C98  4C01                     LDV     $R4,=1                  ANY RANGE
003114  0C99  C380 1102                 LNJ     $B4,<MCCB               MAKE CCB
003115  0C9B  2800                     DC      <RTB                    ANY ADDRESS
003116  0C9C  0060                     DC      X'60'                   LAST BLOCK, VALID
003117                          *
003118  0C9D  B380 0F7D                 LNJ     $B3,<SETLCT             SEND OUT LCT BYTES
003119  0C9F  0CA2                     DC      <SFLG                   FROM HERE
003120  0CA0  0F80 0C68                 B       <INST2
003121                          *
003122  0CA2  0317              SFLG    DC      X'0317'                  SETS LCT X'17' NON ZERO
003123  0CA3  0000                     DC      0                       END OF LIST
003124                          *                                       ON THIS CHANNEL.
003125  0CA4  8AD3              INST3   INC     =$R3
003126  0CA5  0F80 0C12                 B       <INLOOP
003127                          *
003128                          * CHANNEL PROGRAM FOR ABOVE TEST
003129                          *
003130  0CA7  0201              GNBPRG  DC      =Z'0201'                GNB, WAIT
003131  0CA8  9005                     DC      =Z'9005'                LD =5
003132  0CA9  0000                     DC      =Z'0'                   NOP NOP
003133  0CAA  0500                     DC      =Z'0500'                DEC NOP
003134  0CAB  F2FC                     DC      =Z'F2FC'                BZF -3
003135  0CAC  0201                     DC      =Z'0201'                GNB, WAIT
003136  0CAD  0800                     DC      =Z'0800'                INTR,NOP
003137  0CAE  0201                     DC      =Z'0201'                GNB, WAIT
003138  0CAF  0001                     DC      =Z'0001'                NOP, WAIT
003139  0CB0  0000                     DC      0                       SPARE
003140                          *
003141                          *-----------------------------------------------------------
003142                          * * * * * * *
003143                          * DATA SET SCAN TEST
003144                          *
003145  0CB1  9870 5320         DSST    LDR     $R1,=A'S '
003146  0CB3  C380 13E4                 LNJ     $B4,<PLB                REPORT TEST
003147  0CB5  8753                     CL      =$R3
003148  0CB6  C380 105D         DSSTLP  LNJ     $B4,<FLN                FIND ACTIVE LINE
003149  0CB8  8382                     JMP     $B2                     END OF TEST
003150  0CB9  C380 106D                 LNJ     $B4,<GENITZ             INITIALIZE
003151  0CBB  C380 117D                 LNJ     $B4,<DLAY               DELAY 25 MS
003152                          * LOAD CHANNEL PROGRAM FOR DATA SET SCAN
003153  0CBD  BF00 1347                 STR     $R3,<IPR
003154  0CBF  8953                     LBT     =$R3,=X'1'              MAKE SURE WE HAVE A XMIT CHANNEL
        0CC0  0001
003155  0CC1  9380 0FDA                 LNJ     $B1,<SDATA              SEND PROGRAM
003156  0CC3  0DAB                     DC      <SCNPRG                 POINTER TO SCAN PROGRAM
003157  0CC4  0004                     DC      4                       RANGE
003158  0CC5  0200                     DC      =X'200'                 RAM ADDRESS
003159  0CC6  0000                     DC      0                       START ON EVEN BYTE
003160                          *
003161  0CC7  B800 1347                 LDR     $R3,<IPR                GET BACK TEST CHANNEL
003162                          *
003163                          * OUTPUT LCT DATA
003164                          *
003165                          * DATA SCAN MASK = ALL ZERO'S (0)
003166                          * MAKE CLA STATUS = ALL ONES (F)
003167                          *
003168  0CC9  B380 0F7D                 LNJ     $B3,<SETLCT             SET LCT
003169  0CCB  0CCD                     DC      <SCTEST
003170  0CCC  0F90                     B       >SCTST1
003171                          *
003172  0CCD  000F              SCTEST  DC      =Z'000F'                RECEIVE MASK
003173  0CCE  002F                     DC      =Z'002F'                XMIT MASK
003174  0CCF  FF0E                     DC      =Z'FF0E'                RECEIVE CLA STATUS SET TO FF
003175  0CD0  FF2E                     DC      =Z'FF2E'                XMIT CLA STATUS SET TO FF
003176  0CD1  0206              INTLCT  DC      =Z'0206'                RCV P COUNTER,<MSB
003177  0CD2  0007                     DC      =Z'0007'                RCV P COUNTER,<LSB
003178  0CD3  0226                     DC      =Z'0226'                XMIT P COUNTER,<MSB
003179  0CD4  0027                     DC      =Z'0027'                XMIT P COUNTER,<LSB
003180  0CD5  0010                     DC      =Z'0010'                RCV STAT (LCT)
003181  0CD6  0011                     DC      =Z'0011'
003182  0CD7  0030                     DC      =Z'0030'                XMIT STATUS (LCT)
003183  0CD8  0031                     DC      =Z'0031'
003184  0CD9  0008                     DC      =Z'0008'                RCV SCAN CONTROL
003185  0CDA  0028                     DC      =Z'0028'                XMIT SCAN CONTROL
003186  0CDB  0000                     DC      0
003187                          * START SCAN AND CHECK THAT MASK OF 0 INHIBITS ITS OPERATION
003188  0CDC  9870 C008         SCTST1  LDR     $R1,=Z'C008'            DATA SCAN = 1, SET LCT STAT = 1
003189  0CDE  3B02                     BEVN    $R3,>$+2
003190  0CDF  1E20                     ADV     $R1,=X'20'              MODIFY FOR TRANSMIT
003191  0CE0  C380 0DC3                 LNJ     $B4,<OUTLCT             SEND OUT CHANNEL COMMAND BYTE
003192  0CE2  C380 1184                 LNJ     $B4,<DLAYLG             DELAY 250MS
003193  0CE4  C380 0836                 LNJ     $B4,<ILCTST             INPUT LCT STATUS TO R5
003194  0CE6  82D5                     LB      =$R5,=X'10'             GET DATA SET CHANGE BIT
        0CE7  0010
003195  0CE8  0583                     BBF     >$+2+$AF
003196  0CE9  F380 11CB                 LNJ     $B7,<ERROR              DATA SET SCAN BIT SET WITH
003197                          *                                       MASK OF 0
```

```
003198                                  *
003199                                  * READ CLA STATUS BY INPUT DATA SET STATUS
003200  0CEB  9870 000E                        LDR     $R1,=X'E'
003201  0CED  3B02                            BEVN    $R3,>$+2                BRANCH IF RECEIVE CHAN
003202  0CEE  1E20                            ADV     $R1,=X'20'              FORM CLA ADDRESS
003203  0CEF  9F57                            STR     $R1,=$R7
003204  0CF0  C380 0E77                       LNJ     $B4,<DSSTA              READ DATA SET STATUS
003205  0CF2  D570 FF00                       AND     $R5,=Z'FF00'            STRIP OFF UNUSED BYTE
003206  0CF4  D657                            XOR     $R5,=$R7                FORM STATUS WITH ADDRESS
003207  0CF5  DF00 1349                       STR     $R5,<TEMP1
003208  0CF7  8655                            CPL     =$R5                    FORM COMPLEMENT OF STATUS
003209  0CF8  D570 FF00                       AND     $R5,=Z'FF00'            STRIP ADDRESS
003210  0CFA  D657                            XOR     $R5,=$R7                OR IN ADDRESS
003211                                  * OUTPUT COMPLEMENT OF NOMINAL CLA STATUS. DATA SCAN IS STILL
003212                                  * PROGRESSING.
003213  0CFB  DF00 134A                       STR     $R5,<TEMP5T             FIRST STORE AWAY DATA
003214  0CFD  C380 0DC3                       LNJ     $B4,<OUTLCT             OUTPUT BYTE TO LCT
003215  0CFF  C3C0 047D                       LNJ     $B4,DLAY                WAIT FOR FIRMWARE TO FINISH
003216                                  *
003217                                  * INPUT LCT STATUS AND CHECK DATA SET CHANGE IS RESET
003218                                  *
003219  0D01  C380 0836                       LNJ     $B4,<ILCTST             INPUT LCT STATUS TO R5
003220  0D03  82D5                            LB      =$R5,=X'10'             GET DATA SET CHANGE BIT
      0D04  0010
003221  0D05  0563                            BBF     >$+2+$AF
003222  0D06  F380 11CB                       LNJ     $B7,<ERROR              DATA SET SCAN BIT SET WITH
003223                                  *                                     MASK OF 0
003224                                  *
003225                                  * START OF LOOP WHICH WILL TEST THE DATA SET MASK
003226                                  *
003227  0D08  9870 0400                       LDR     $R1,=X'400'
003228  0D0A  9F00 135E                       STR     $R1,<MASK
003229                                  *
003230                                  *
003231                                  *
003232  0D0C  B3C0 0270               DSLOOP  LNJ     $B3,SETLCT              SEND OUT LCT
003233  0D0E  0CCD                            DC      <SCIEST
003234  0D0F  9800 135E                       LDR     $R1,<MASK
003235  0D11  1001                            SOL     $R1,1
003236  0D12  0600 0D52                       BCT     <SCI11A                 EXIT IF NO MORE BITS TO TEST
003237  0D14  9F00 135E                       STR     $R1,<MASK               STORE MASK
003238  0D16  9670 000F                       XOR     $R1,=X'F'               OR IN ADDRESS FOR RCV MASK
003239  0D18  3B02                            BEVN    $R3,>$+2
003240  0D19  1E20                            ADV     $R1,=X'20'              MODIFY IF XMIT
003241  0D1A  C380 0DC3                       LNJ     $B4,<OUTLCT             OUTPUT MASK (R1) TO LCT
003242                                  * OUTPUT BYTE TO CLA POSISTION IN LCT. THIS BYTE WILL TEST
003243                                  * ONE BIT POSISTION CORRESPONDING TO THE MASK WHICH HAS ONE BIT SET.
003244                                  * EACH BIT POSISTION IN THE STATUS MASK IS CHECKED ONCE
003245                                  *
003246  0D1C  C380 117D                       LNJ     $B4,<DLAY               DELAY 25 MSEC
003247  0D1E  9800 1349                       LDR     $R1,<TEMP1              GET CLA NORMAL STAT + ADDRESS
003248                                  *                                     WITH LCT ADDRESS
003249  0D20  9600 135E                       XOR     $R1,<MASK               COMPLEMENT BIT TO TEST
003250  0D22  C380 0DC3                       LNJ     $B4,<OUTLCT             OUTPUT TO CLA STATUS
003251                                  * SET UP FOR ERROR INTERRUPT
003252  0D24  C380 0DAD                       LNJ     $B4,<LEVRCH             SET LCT BYTES FOR LEVEL = 6 + RETURN CHAN
003253  0D26  9380 0EC9                       LNJ     $B1,<INTITZ             INITIALIZE INTERRUPTS
003254  0D28  1C06                            LDV     $R1,=6
003255  0D29  9F30 26F6                       STR     $R1,<ILV,$R3            SET FOR LEVEL 6 IN LEVEL TABLE
003256                                  *
003257                                  * SWITCH TO LEVEL 63
003258                                  * DATA SET SCAN WILL TURN ON WITH CONTROL = "D0"
003259  0D2B  9B80 0D38                       LAB     $B1,<LV63TB
003260  0D2D  9F80 0E9D                       STB     $B1,<L63SV2             SET PVECTOR IN LEVEL 63 SAVE
003261  0D2F  C380 11BB                       LNJ     $B4,<RTC                SET RTC TO INTERRUPT
003262  0D31  0006                            DC      6                       AFTER 50 MSEC
003263  0D32  8F00 0E9F                       SAVE    <L63SV3,=X'7884'        PASS ALONG REGISTERS
      0D34  7884
003264  0D35  8E70 803F                       LEV     =Z'803F'                SUSPEND CURRENT AND GO TO 63
003265  0D37  0F66                            B       >DOL2                   RETURN HERE AFTER R7C RUPT
003266                                  *
003267                                  * LEVEL 63 CODE. OUTPUT A DATA SET SCAN CONTROL WORD
003268                                  * WITH BIT 0 = 1 (SCAN), BIT 1 =1 (SET STATUS), BIT 2 = 0
003269                                  * (INTERRUPT),<BIT 3 = 1 (START CHANNEL PROGRAM)
003270                                  *
003271  0D38  B380 0F7D               LV63TB  LNJ     $B3,<SETLCT             OUTPUT DATA SCAN BYTE
003272  0D3A  0DC0                            DC      <SCISVA
003273  0D3B  0F02                    NOER20  NOP     >DOL2                   WAIT FOR RTC INTERRUPT
003274  0D3C  0FFF                            B       >NOER20
003275                                  *
003276                                  * RETURN HERE VIA RTC INTERRUPT. THEN INPUT LCT STATUS
003277                                  *
003278  0D3D  0005                    DOL2    RTCF                            TURN OFF REAL TIME CLOCK
003279  0D3E  C380 0836                       LNJ     $B4,<ILCTST             INPUT LCT STATUS TO R5
003280  0D40  82D5                            LB      =$R5,=X'10'             GET DATA SET CHANGE BIT
      0D41  0010
003281  0D42  0503                            BBT     >$+2+$AF
003282  0D43  F380 11CB                       LNJ     $B7,<ERROR              DATA SET SCAN BIT NOT SET
003283                                  *
003284                                  * RESET LEVEL TO ZERO SO BLOCK READ WON'T CAUSE INTERRUPT.
003285                                  *
003286  0D45  1C0D                            LDV     $R1,=X'D'               ADDRESS OF LEVEL
003287  0D46  C380 0DC3                       LNJ     $B4,<OUTLCT             OUTPUT 0 TO LEVEL
003288                                  *
003289  0D48  C380 0DDC                       LNJ     $B4,<RPVLU              READ P VALUE TO TEMP
003290  0D4A  E870 0202                       LDR     $R6,=X'202'
003291  0D4C  D956                            CMR     $R5,=$R6
003292  0D4D  0903                            BE      >$+2+$AF
003293  0D4E  F380 11CB                       LNJ     $B7,<ERROR              CHANNEL PROGRAM NOT STARTED BY
003294                                  *                                     SCAN WITH BIT 3 OF CHAN COMMAND = 1.
003295  0D50  0F80 0D0C                       B       <DSLOOP                 TEST NEXT MASK BIT
003296                                  *
003297                                  * TEST THAT DATA SET SCAN WILL CAUSE INTERRUPT
003298                                  *
003299  0D52  C380 0DCF               SCI11A  LNJ     $B4,<CHINTZ             INITIALIZE CHANNEL
003300  0D54  C380 117D                       LNJ     $B4,<DLAY               DELAY 25 MS.
003301                                  *
003302  0D56  B3C0 0226                       LNJ     $B3,SETLCT              SEND OUT LCT
003303  0D58  0CCD                            DC      <SCIEST
003304                                  *
003305                                  * OUTPUT LCT DATA
003306                                  *
003307                                  *       MASK = ALL ONES (F)
```

```
003308                          *
003309  0D59  B380 0F7D                LNJ   $B3,<SETLCT          SET LCI
003310  0D5B  0D5D                    DC    <SCIT1               TABLE ADDRESS
003311  0D5C  0F84                    B     >SCIT2
003312                          *
003313  0D5D  FF0F              SCIT1 DC    =Z'FF0F'             RECEIVE MASK
003314  0D5E  FF2F                    DC    =Z'FF2F'             XMIT MASK
003315  0D5F  0000                    DC    0                    END OF TABLE
003316  0D60  C380 0DAD         SCIT2 LNJ   $B4,<LEVRCH          SET LCI BYTES FOR LEV6 + REI. CHANNEL
003317  0D62  9380 0EC9               LNJ   $B1,<INTITZ          INITIALIZE INTERRUPTS
003318  0D64  1C06                    LDV   $R1,=6
003319  0D65  9F30 26F6               STR   $R1,<ILV.$R3         SET FOR LEVEL 6 IN LEVEL TABLE
003320  0D67  C380 104C               LNJ   $B4,<SETVEC          SET UP VECTOR FOR LEVEL
003321  0D69  1CFF                    LDV   $R1,=-1
003322  0D6A  9F30 26D6               STR   $R1,<INMB.$R3        SET FOR ONE RUPT EXPECTED
003323  0D6C  1C0F                    LDV   $R1,=X'F'            SET FLAG TO SCAN
003324  0D6D  9F00 134B               STR   $R1,<SCFLG
003325                          *
003326                          * SEND OUT COMPLEMENT OF NORMAL STATUS
003327                          *
003328  0D6F  9800 134A                LDR   $R1,<TEMPST
003329  0D71  C380 0DC3               LNJ   $B4,<OUTLCI          OUT PUT VALUE
003330                          *
003331                          * SWITCH TO LEVEL 63
003332                          *
003333  0D73  9B80 0D80                LAB   $B1,<SCIT3
003334  0D75  9F80 0E9D               STB   $B1,<L63SV2          SET P VECTOR IN LEVEL 63 SAVE
003335  0D77  C380 11BB               LNJ   $B4,<RTC             SET RTC TO INTERRUPT
003336  0D79  0006                    DC    6                    AFTER 50 MS
003337  0D7A  8F00 0E9F               SAVE  <L63SV3,=X'7884'
        0D7C  7884
003338  0D7D  8E70 803F               LEV   =Z'803F'             SUSPEND CURRENT AND GO TO 63
003339  0D7F  0F8B                    B     >DOL3
003340
003341                          * LEVEL 63 CODE. OUTPUT A DATA SCAN CONTROL WORD WITH
003342                          * BIT 0=1, SCAN; BIT 1=0 (SET STATUS) BIT 2=1 (INTERRUPT),
003343                          * BIT 3=0 (START CHANNEL PROGRAM).
003344  0D80  9870 A008         SCIT3 LDR   $R1,=Z'A008'
003345  0D82  3B02                    BEVN  $R3,>$+2             BRANCH IF RECEIVE
003346  0D83  1E20                    ADV   $R1,=X'20'           MODIFY FOR XMIT
003347  0D84  C380 0DC3               LNJ   $B4,<OUTLCT          OUTPUT BYTE
003348  0D86  0F00 0D80         WRUPT NOP   <SCIT3
003349  0D88  0F58                    NOP   >SCIT2
003350  0D89  0FFD                    B     >WRUPT               WAIT FOR INTERRUPT
003351                          *
003352                          * RETURN HERE VIA RTC INTERRUPT. CHECK LCI STATUS TO INSURE
003353                          * THAT DATA SET SCAN BIT IS RESET.
003354                          *
003355  0D8A  0005              DOL3  RTCF                       TURN OFF REAL TIME CLOCK
003356  0D8B  C3C0 03F1               LNJ   $B4,DLAY             WAIT FOR CHANNEL PROGRAM TO FINISH
003357  0D8D  C380 0836               LNJ   $B4,<ILCTSI          INPUT LCI STATUS TO R5
003358  0D8F  82D5                    LB    =$R5,=X'10'          GET DATA SET CHANGE BIT
        0D90  0010
003359  0D91  0583                    BBF   >$+2+$AF
003360  0D92  F380 11CB               LNJ   $B7,<ERROR           DATA SET SCAN BIT IS SET WHEN
003361                          *                                IT SHOULDN'T BE.
003362  0D94  82D5                    LB    =$R5,=X'4000'        GET INTERRUPT STATUS BIT
        0D95  4000
003363  0D96  0503                    BBT   >$+2+$AF
003364  0D97  F380 11CB               LNJ   $B7,<ERROR           DATA SET SCAN DIDN'T SET RUPT BIT
003365                          *
003366                          * RESET LEVEL IN LCT SO FOLLOWING BLOCK READ WILL NOT CAUSE AN INTERRUPT
003367                          *
003368  0D99  1C0D                    LDV   $R1,=X'D'            ADDRESS OF LEVEL
003369  0D9A  3B02                    BEVN  $R3,>$+2             BRANCH IF RCV
003370  0D9B  1E20                    ADV   $R1,=X'20'
003371  0D9C  C380 0DC3               LNJ   $B4,<OUTLCT          OUTPUT 0 TO LEVEL
003372                          * CHECK P COUNTER TO INSURE THAT THE CHANNEL PROGRAM
003373                          * DIDN'T START.
003374                          *
003375  0D9E  C380 0DDC               LNJ   $B4,<RPVLU           READ P VALUE TO TEMP
003376  0DA0  E870 0200               LDR   $R6,=X'200'
003377  0DA2  D956                    CMR   $R5,=$R6
003378  0DA3  0903                    BE    >$+2+$AF
003379  0DA4  F380 11CB               LNJ   $B7,<ERROR           CHANNEL PROGRAM  STARTED
003380                          *                                BY DATA SET SCAN  WHEN IT
003381                          *                                SHOULD NOT HAVE.
003382                          *
003383                          * CHECK THAT INTERRUPT OCCURED
003384                          *
003385  0DA6  9380 0F6A               LNJ   $B1,<CHLV            CHECK INTERRUPT OCCURED ON CHANNEL
003386                          *
003387                          * TEST FOR NEXT CHANNEL
003388                          *
003389  0DA8  8AD3                    INC   =$R3
003390  0DA9  0F80 0CB6               B     <DSSTLP              GO TO NEXT CHANNEL
003391                          *
003392                          * CHANNEL PROGRAM FOR SCAN TEST
003393                          *
003394  0DAB  0001              SCNPRG DC    =Z'0001'             NOP,<WAIT
003395  0DAC  0101                    DC    =Z'0101'             WAIT, WAIT
003396                          *
003397                          * SUBROUTINES FOR INTERRUPT TESTS
003398  0DAD  CED5              LEVRCH SWB   $B4,=$B5             SAVE B4
003399  0DAE  8C51                    STS   =$R1                 GET S REG
003400  0DAF  9570 03C0               AND   $R1,=Z'03C0'         STRIP TO CPU ID
003401  0DB1  9F00 13B5               STR   $R1,<RTCHN
003402  0DB3  9670 0006               XOR   $R1,=6               PUT IN LEVEL 6
003403  0DB5  C800 13C5               LDR   $R4,<CONT11          FUNCTION CODE FOR OUTPUT INT CONT
003404  0DB7  C380 10EC               LNJ   $B4,<CGSCH           MODIFY FOR CHANNEL
003405  0DB9  8051                    IO    =$R1,=$R4            OUTPUT INTERRUPT CONTROL
        0DBA  0054
003406  0DBB  0703                    BIOT  >$+2+$AF
003407  0DBC  F380 11CB               LNJ   $B7,<ERROR           COMMAND WAS NAK'ED
003408  0DBE  CED5                    SWB   $B4,=$B5             RESTORE B4
003409  0DBF  8384                    JMP   $B4
003410                          * LCT CONTROL FOR SCAN TEST.
003411                          *
003412  0DC0  D008              SCTSVA DC    =Z'D008'             RCV SCAN CONTROL
003413  0DC1  D028                    DC    =Z'D028'             =ZMIT SCAN CONTROL
003414  0DC2  0000                    DC    0
003415                          *
003416                          * OUTPUT LCT BYTE CONTAINED IN R1
```

```
003417  0DC3  CED5                OUTLCT SWB    $B4,=$B5
003418  0DC4  C800 13C0                  LDR    $R4,<CONT5          OUTPUT LCT BYTE FUNC CODE
003419  0DC6  C380 10EC                  LNJ    $B4,<CGSCH          MODIFY FOR FOR CHANNEL
003420  0DC8  8051                      IO     =$R1,=$R4           OUTPUT BYTE
        0DC9  0054
003421  0DCA  0703                      BIOT   >$+2+$AF
003422  0DCB  F380 11CB                 LNJ    $B7,<ERROR          COMMAND WAS NAK'ED
003423  0DCD  CED5                      SWB    $B4,=$B5            RESTORE B4
003424  0DCE  8384                      JMP    $B4
003425                            *
003426                            * ISSUE CHANNEL INITIALIZE
003427  0DCF  CED5                CHINTZ SWB    $B4,=$B5
003428  0DD0  C800 13C2                  LDR    $R4,<CONT7          CHANNEL CONTROL
003429  0DD2  C380 10EC                  LNJ    $B4,<CGSCH          MODIFY FOR CHANNEL
003430  0DD4  8070 8000                  IO     =Z'8000',=$R4
        0DD6  0054
003431  0DD7  0703                      BIOT   >$+2+$AF
003432  0DD8  F380 11CB                 LNJ    $B7,<ERROR          COMMAND WAS NAKED
003433  0DDA  CED5                      SWB    $B4,=$B5            RESTORE B5
003434  0DDB  8384                      JMP    $B4
003435                            *
003436                            * READ P VALUE FROM RAM
003437                            *
003438  0DDC  1C06                RPVLU  LDV    $R1,=6
003439  0DDD  BF00 1347                  STR    $R3,<IPR
003440  0DDF  3B03                      BEVN   $R3,>DOL4           BRANCH IF READ
003441  0DE0  8883                      DEC    =$R3                GET READ CHANNEL
003442  0DE1  1C26                      LDV    $R1,=38
003443  0DE2  C853                DOL4   LDR    $R4,=$R3
003444  0DE3  4F20                      MLV    $R4,=X'20'          FORM START OF LCT AREA
003445  0DE4  9A54                      ADD    $R1,=$R4            FORM ADDRESS OF P COUNTER
003446  0DE5  9F00 0DEB                 STR    $R1,<PVLU1
003447  0DE7  9380 1009                  LNJ    $B1,<RDATA          DO BLOCK READ
003448  0DE9  1348                      DC     <TEMP               TO HERE
003449  0DEA  0002                      DC     2                   2 BYTES
003450  0DEB  0000                PVLU1  DC     0                   RAM ADDRESS
003451  0DEC  0000                      DC     0                   START AT EVEN CPU BYTE
003452                            *
003453  0DED  D800 1348                  LDR    $R5,<TEMP
003454  0DEF  D570 0FFF                  AND    $R5,=Z'0FFF'
003455  0DF1  DF00 1348                  STR    $R5,<TEMP
003456  0DF3  B800 1347                  LDR    $R3,<IPR            GET BACK TESTED CHANNEL
003457  0DF5  8384                      JMP    $B4
003458                            *---------------------------------------------------------------
003459                            *
003460                            * 2ND INTERRUPT TEST.
003461                            *     DO BLOCK WRITES AND READS AND CHECK INTERRUPTS.
003462                            *     3 DEFERED INTERRUPTS ARE ACCUMULATED ON ALL CHANNELS
003463                            *      - LEVELS 5 - 62 ARE USED
003464                            *
003465                            *
003466  0DF6  9870 5420           SITST  LDR    $R1,=A'1 '
003467  0DF8  C380 13E4                  LNJ    $B4,<PLB            REPORT TEST
003468  0DFA  8E70 0080                  LEV    =Z'0080'            SUSPEND + QUICK CHANGE TO LEV 0
003469  0DFC  5C05                      LDV    $R5,=5              STARTING LEVEL NUMBER
003470  0DFD  DF00 0E2F                 STR    $R5,<LVSTR          STARTING LEVEL
003471                            *
003472  0DFF  BB80 2400           PRILP  LAB    $B3,<SDB            GET ADDRESS OF SEND DATA BUFFER
003473  0E01  BF80 0E15                  STB    $B3,<ADD
003474  0E03  C870 0200                  LDR    $R4,=X'200'         RAM ADDRESS
003475  0E05  CF00 0E16                  STR    $R4,<RAD
003476  0E07  8753                      CL     =$R3                R3 IS CHANNEL COUNTER
003477  0E08  2CF0                      LDV    $R2,=-16
003478  0E09  4C02                      LDV    $R4,=2              RANGE OF TWO BYTES
003479                            *
003480                            * INITIALIZE CONTROLLER
003481                            *
003482  0E0A  C380 106D                  LNJ    $B4,<GENITZ         MLCC GENERAL INITIALIZE
003483  0E0C  9380 0EC9                  LNJ    $B1,<INTITZ         INITIALIZE INTERRUPTS
003484                            *
003485                            * SET UP 3 CCB'S FOR EACH CHANNEL
003486                            *
003487  0E0E  C380 105D           CCBLP  LNJ    $B4,<FLN            FIND ACTIVE LINE
003488  0E10  0F92                      B      >RT1                ALL HAVE BEEN FOUND
003489                            *
003490  0E11  8751                      CL     =$R1                R1= 0 MEANS LEFT BYTE BOUNDRY
003491  0E12  5CFD                      LDV    $R5,=-3
003492                            *
003493  0E13  C380 1102           NEXT5  LNJ    $B4,<MCCB           FORM CCB
003494  0E15  1341                ADD    DC     <DUMMY
003495  0E16  0000                RAD    DC     =X'0'               RAM ADDRESS
003496                            *
003497  0E17  57FC                      BINC   $R5,>NEXT5
003498  0E18  8AD3                      INC    =$R3                SET FOR NEXT CHANNEL
003499  0E19  8A80 0E15                 INC    <ADD+$AF-1          BUMP CPU ADDRESS
003500  0E1B  8A80 0E16                 INC    <RAD                BUMP RAM ADDRESS
003501  0E1D  8A80 0E16                 INC    <RAD                BUMP AGAIN
003502  0E1F  AF48 FFF5                 STR    $R2,*ADD            SET UP DATA - ASCENDING PATTERN, -16 TO -1
003503  0E21  27ED                      BINC   $R2,>CCBLP          BRANCH IF MORE TO GO
003504                            * SET UP LEVELS
003505  0E22  D800 0E2F           RT1    LDR    $R5,<LVSTR          SET LEVEL START
003506  0E24  A870 8000                  LDR    $R2,=Z'8000'
003507  0E26  9852                RT6    LDR    $R1,=$R2
003508  0E27  9500 1339                  AND    $R1,<IMASK          STRIP TO CHANNEL BIT
003509  0E29  1900 0E37                  BEZ    $R1,<RT7A
003510                            *
003511  0E2B  9F00 0E30                  STR    $R1,<CHAN           INITIALIZE CHANNEL
003512  0E2D  9380 0F59                  LNJ    $B1,<FTBL           SET LEVEL
003513  0E2F  0000                LVSTR  DC     0                   LEVEL
003514  0E30  0000                CHAN   DC     0                   CHANNEL
003515  0E31  26F6                      DC     <ILV                LEVEL TABLE
003516  0E32  8AD5                      INC    =$R5
003517  0E33  5D3F                      CMV    $R5,=63             CHECK IF LAST LEVEL
003518  0E34  0905                      BE     >RT5                BRANCH IF LAST LEVEL
003519  0E35  DF00 0E2F                 STR    $R5,<LVSTR          SET FOR NEXT LEVEL
003520  0E37  2041                RT7A   SOR    $R2,1
003521  0E38  29EE                      BNEZ   $R2,>RT6            DO NEXT CHANNEL
003522                            *
003523                            *  FORM MASK FOR LINES TO BE TESTED
003524                            *
003525  0E39  1CF0                RT5    LDV    $R1,=-16
003526  0E3A  8752                      CL     =$R2
003527                            *
```

```
003528  0E3B  B810 2706          RT7     LDR     $R3,<ILV+16,$R1       GET LEVEL FOR CHANNEL
003529  0E3D  8AD1                      INC     =$R1
003530  0E3E  2011                      SCL     $R2,1
003531  0E3F  3903                      BEZ     $R3,>RT8
003532  0E40  A670 0001                  XOR     $R2,=Z'0001'          PUT IN BIT FOR CHANNEL
003533  0E42  1879                RT8     BLZ     $R1,>RT7              MORE TO GO
003534                             *
003535  0E43  AF00 0E4C                  STR     $R2,<RT2
003536  0E45  AF00 0E51                  STR     $R2,<RT3
003537  0E47  AF00 135E                  STR     $R2,<MASK
003538                             *
003539                             *
003540                             * SET UP INTERRUPT INFORMATION
003541                             *
003542  0E49  9380 0F59                  LNJ     $B1,<FTBL             SET EXPECTED INTERRUPT COUNT
003543  0E4B  FFFD                       DC      -3                    3 INTERRUPT/CHANNEL
003544  0E4C  0000                RT2     DC      0                     MASK FOR TESTED CHANNELS
003545  0E4D  26D6                       DC      <INMB                 TABLE ADDRESS
003546                             *
003547  0E4E  9380 0F59                  LNJ     $B1,<FTBL             SET EXPECTED STATUS
003548  0E50  5000                       DC      =Z'5000'              WHAT STAT SHOULD BE
003549  0E51  0000                RT3     DC      0                     MASK FOR TESTED CHANNELS
003550  0E52  26E6                       DC      <IST                  STATUS TABLE
003551                             *
003552                             * SET INTERRUPT VECTORS
003553                             *
003554  0E53  C380 104C                  LNJ     $B4,<SETVEC
003555                             *
003556                             * SEND OUT LEVEL AND CHANNEL
003557                             *
003558  0E55  B380 103B                  LNJ     $B3,<SETV
003559  0E57  8753                       CL      =$R3
003560  0E58  C380 105D                  LNJ     $B4,<FLN              FIND LINE NUMBER
003561  0E5A  0000                       HLT                           NO ACTIVE LINE NUMBERS
003562                             *
003563                             * TURN ON ALL CHANNELS
003564                             *
003565  0E5B  1CFD                       LDV     $R1,=-3               3 BLOCK OPERATIONS/CHAN
003566  0E5C  B380 1021            RT10    LNJ     $B3,<TRNON            TURN ON ALL CHANNELS TO BE TESTED
003567  0E5E  0400                       DC      =Z'0400'              START BLOCK WRITE CHANNEL CONTROL
003568  0E5F  0001                       DC      1                     BLOCK WRITE OPERATION
003569                             *
003570  0E60  B380 1021                  LNJ     $B3,<TRNON            TURN ON CHANNELS
003571  0E62  0800                       DC      =Z'0800'              BLOCK READ OPERATION
003572  0E63  0002                       DC      2                     BLOCK READ
003573                             *
003574  0E64  C380 117D                  LNJ     $B4,<DLAY             DELAY 25 MS
003575  0E66  1780 0E5C                  BINC    $R1,<RT10             BRANCH IF NOT LAST TIME
003576                             *
003577                             * SET UP RTC FOR 100 MILLISEC
003578                             *
003579  0E68  C380 11BB                  LNJ     $B4,<RTC
003580  0E6A  000C                       DC      =12                   120'S OF A SEC
003581                             * NOW SWITCH LEVELS TO 63 AND WAIT FOR SMOKE TO CLEAR
003582  0E6B  8F00 0E9F                  SAVE    <L63SV3,=X'7884'
        0E6D  7884
003583  0E6E  8E70 803F                  LEV     =Z'803F'              SUSPEND CURRENT AND DISPATCH 63
003584                             *
003585                             * RETURN HERE AFTER INTERRUPTS HAVE OCCURED
003586  0E70  0005                       RTCF                          SHUT OFF RTC
003587                             * CHECK ALL INTERRUPTS OCCURED
003588  0E71  9380 0F6A                  LNJ     $B1,<CHLV             CHECK ALL LEVELS INTERRUPTED
003589                             *
003590                             * END OF 1 LOOP THRU
003591                             *
003592  0E73  5D3F                       CMV     $R5,=63               CHECK FOR LAST LEVEL
003593  0E74  0200 0DFF                  BL      <PRILP                MORE TO GO
003594  0E76  8382                       JMP     $B2                   EXIT TEST
003595                             *
003596                             * READ DATA SET STATUS
003597                             *
003598  0E77  8F40 050C            DSSTA   SAVE    SAV3,=Z'0008'         SAVE B4
        0E79  0008
003599  0E7A  C800 13C6                  LDR     $R4,<CONT12           GET FUNCTION CODE
003600  0E7C  C380 10EC                  LNJ     $B4,<CGSCH            OR IN CHANNEL NUMBER
003601  0E7E  8055                       IO      =$R5,=$R4             INPUT DATA SET STATUS
        0E7F  0054
003602  0E80  0703                       BIOT    >$+2+$AF
003603  0E81  F380 11CB                  LNJ     $B7,<ERROR            COMMAND WAS NAK'ED
003604  0E83  8FC0 0500                  RSTR    SAV3,=Z'0008'
        0E85  0008
003605  0E86  8384                       JMP     $B4                   DONE
003606                             *
003607                             *
003608                             *
003609                             *
003610                             *
003611                             * INTERRUPT SAVE AREA - LEV 4
003612                             *
003613  0E87  0000                 ISAV4   RESV    $AF,0                 TRAP SAVE POINTER
003614  0E88  0000                         RESV    1,0                   CHAN, LEVEL
003615  0E89  0000                         RESV    1,X'0'                SAVE NO REGISTERS
003616  0E8A  0000                         RESV    1,0                   RFU
003617  0E8B  0FCD                 IS4S    DC      <IMON                 CCB SCHEDULAR
003618  0E8C  FFFF                         DC      -1                    SET PRIVILEGE BIT
003619                             *
003620                             * INTERRUPT SAVE AREA - ERROR INTERRUPT
003621                             *
003622  0E8D  0000                 ERRINT  RESV    $AF,0                 TRAP SAVE AREA POINTER
003623  0E8E  0000                         RESV    1,0                   DEVICE ID
003624  0E8F  0000                         RESV    1,0                   RFU
003625  0E90  0000                         DC      Z'0000'               SAVE
003626  0E91  0EAF                         DC      <ERUPT                ERROR RUPT ROUTINE
003627  0E92  FFFF                         DC      -1                    SET PRIVILEGE
003628                             *
003629                             * RTC SAVE AREA
003630                             *
003631  0E93  0000                 RTCSAV  RESV    $AF,0
003632  0E94  0000                         RESV    1,0                   DEVICE ID GETS STORED HERE
003633  0E95  0000                         RESV    1,0                   RFU
003634  0E96  0000                         DC      Z'0000'               SAVE
003635  0E97  0EB9                 RTCSV1  DC      <RTCHDL               POINTS TO INTERRUPT ROUTINE
003636  0E98  FFFF                         DC      -1                    SET PRIVILEGE BIT
```

```
003637                                  *
003638                                  * INTERRUPT SAVE AREA - LEVEL 63
003639                                  *
003640  0E99  0000                     L63SV1 RESV   $AF,0                    TRAP SAVE POINTER
003641  0E9A  0000                            RESV   1,0                      CHAN,<LEVEL
003642  0E9B  7884                            RESV   1,X'7884'                SAVE R1,R2,R3,R4,I,B5
003643  0E9C  0000                            RESV   1,0                      RFU
003644  0E9D  0EA5                     L63SV2 DC     <L63HDL                 POINTS TO INT. ROUTINE
003645  0E9E  FFFF                            DC     -1                      SET PRIVILEGE BIT
003646  0E9F                           L63SV3 RESV   5+$AF                   SAVE AREA
003647                                  *
003648                                  *
003649                                  *
003650                                  * LEVEL 63 CONTAINS 'WAIT' LOOP
003651                                  *
003652  0EA5  0F00 1341                 L63HDL NOP    <DUMMY
003653  0EA7  0FFE                            B      >L63HDL
003654                                  *
003655                                  *  TRAP SAVE AREA IS AT LOC 2 - 9
003656                                  *
003657                                  *
003658                                  * TRAP ERROR ROUTINE
003659                                  *
003660  0EA8  8C00 0EB6                 TRPERR STS    <LAST                   STORE STATUS
003661  0EAA  8E70 0080                        LEV    =X'0080'                ERROR, SHOULD BE NO TRAPS
003662  0EAC  F380 11CB                        LNJ    $B7,<ERROR
003663  0EAE  0000                             HLT
003664                                  *
003665                                  *
003666                                  * ERROR INTERRUPT
003667  0EAF  8C00 0EB6                 ERUPT  STS    <LAST                   STORE LAST STATUS
003668  0EB1  8E70 0080                        LEV    =X'0080'                ERROR, BAD INTERRUPT OCCURED
003669  0EB3  F380 11CB                        LNJ    $B7,<ERROR
003670  0EB5  0000                             HLT
003671  0EB6  0000                     LAST   DC     0                       LAST STATUS STORED HERE
003672                                  *
003673                                  * RTC INTERRUPT ROUTINE
003674                                  *
003675  0EB7  8E70 8000                 SETNEW LEV    =Z'8000'                SCHEDULE LEVEL = 0
003676  0EB9  8A80 0EBD                 RTCHDL INC    <RTCFLG
003677  0EBB  83C0 FFFB                        JMP    SETNEW                  GO TO LEVEL 0
003678  0EBD  0000                     RTCFLG DC     0                       REAL TIME CLOCK FLAG
003679                                  *
003680                                  * INITIALIZATION TABLE FOR INTERRUPTS
003681                                  *
003682  0EBE  0E97                     INTBL  DC     <RTCSV1                 RTC SAVE POINTER
003683  0EBF  0EB9                            DC     <RTCHDL                 RTC RUPT POINTER
003684  0EC0  0E9D                            DC     <L63SV2                 LEVEL 63 P SAVE
003685  0EC1  0EA5                            DC     <L63HDL                 LEVEL 63 RUPT HANDLER
003686  0EC2  0E8B                            DC     <ISAS                   LEVEL 4 SAVE
003687  0EC3  0FCD                            DC     <IMON
003688  0EC4  0EC8                            DC     <STKPTR                 PRIORITY STACK POINTER
003689  0EC5  2706                            DC     <PRIST
003690  0EC6  0EC7                     INLLST DC     <INLLST+$AF             LAST IN LIST
003691  0EC7  2716                     PRIEND DC     <PRIST+16
003692  0EC8  2706                     STKPTR DC     <PRIST
003693                                  *
003694                                  *
003695                                  *
003696                                  *
003697                                  * INITIALIZATION FOR INTERRUPTS
003698                                  *
003699                                  *    ENTERED BY LNJ $B1,<INTITZ
003700                                  *
003701  0EC9  8F40 049A                 INTITZ SAVE   SAV1,=Z'FFFF'           SAVE ALL
      0ECB  FFFF
003702  0ECC  DB80 0EBE                        LAB    $B5,<INTBL              GET ADDRESS OF LIST
003703  0ECE  ACF5                     ITM1   LDB    $B2,*$B5                B2 GETS POINTER
003704  0ECF  DD80 0EC6                        CMB    $B5,<INLLST
003705  0ED1  0904                            BE     >ININX                  BRANCH MEANS WE GOT END OF LIST
003706  0ED2  BCF5                             LDB    $B3,+$B5                B3 GETS VECTOR
003707  0ED3  BF82                             STB    $B3,$B2                 PUT IN VECTOR
003708  0ED4  0FFA                             B      >ITM1                   DO NEXT IN LIST
003709                                  *
003710  0ED5  DB80 0E8E                 ININX  LAB    $B5,<ERRINT+$AF         GET ADDRESS OF ERROR SAVE
003711  0ED7  AB80 0000          X             LAB    $B2,<ZHISAZ             WHERE TO START STORING IT
003712  0ED9  1CC1                             LDV    $R1,=-63                63 VECTORS
003713  0EDA  0FF2                     ITM2   STB    $B5,+$B2
003714  0EDB  17FF                             BINC   $R1,>ITM2
003715                                  *
003716                                  * SET UP VECTORS FOR TRAPS
003717                                  *
003718  0EDC  1CEC                             LDV    $R1,=-20
003719  0EDD  DB80 0000          X             LAB    $B5,<ZHPTR              CLEAR  TRAP HANDLER POINTERS
003720                                  *
003721  0EDF  AB80 0000          X             LAB    $B2,<ZHISAZ
003722  0EE1  0FE2                     SVEC   STB    $B5,-$B2
003723  0EE2  17FF                             BINC   $R1,>SVEC
003724  0EE3  AB80 0000          X             LAB    $B2,<ZHTSA              GET LOCATION OF TRAP SAVE AREA
003725  0EE5  AF80 0000          X             STB    $B2,<ZHNTSA
003726  0EE7  AB80 0000          X             LAB    $B2,<ZHISAZ             GET RUPT POINTER BASE
003727  0EE9  DB80 26C1                        LAB    $B5,<TOPSAV             ADDRESS OF LEVEL 0 SAVE AREA
003728  0EEB  0F82                             STB    $B5,$B2
003729                                  * PUT IN RTC, LEV 4 + 63 VECTORS
003730  0EEC  DB80 0E94                        LAB    $B5,<RTCSAV+$AF         ADDRESS OF RTC SAVE AREA
003731  0EEE  0FC2  0003                       STB    $B5,$B2.3*$AF
003732  0EF0  DB80 0E88                        LAB    $B5,<ISAV4+$AF          GET LOC OF RUPT SCHEDULAR SAVE
003733  0EF2  0FC2  0004                       STB    $B5,$B2.4*$AF
003734  0EF4  DB80 0E9A                        LAB    $B5,<L63SV1+$AF         GET ADDRESS OF LEV 63 SAVE AREA
003735  0EF6  0FC2  003F                       STB    $B5,$B2.63*$AF
003736  0EF8  8751                             CL     =$R1
003737  0EF9  9F00 0EBD                        STR    $R1,<RTCFLG             CLEAR OUT RTC FLAG
003738  0EFB  9F00 134C                        STR    $R1,<CCBFLG             RESET CCB FLAG
003739  0EFD  9F00 134B                        STR    $R1,<SCFLG              RESET SCAN FLAG
003740  0EFF  9F00 0001          X             STR    $R1,<ZHIAFB+1
003741  0F01  9F00 0002          X             STR    $R1,<ZHIAFB+2
003742  0F03  9F00 0003          X             STR    $R1,<ZHIAFB+3
003743  0F05  9870 8000                        LDR    $R1,=Z'8000'            SET LEVEL FOR 0
003744  0F07  9F00 0000          X             STR    $R1,<ZHIAFB
003745                                  * CLEAR OUT TRAP SAVE AREA
003746  0F09  DB80 1358                        LAB    $B5,<ISA2               GET TRAP SAVE POINTER
003747  0F0B  0F80 0000          X             STB    $B5,<ZHISA
003748                                  *
```

```
003749                                  * INITIALIZE STATUS TABLE
003750                                  *
003751  0F0D  8700 135F                         CL      <MASK1
003752                                          CALL    ZV$F,IST,MASK1,C16
        0F0F  FBC0 0003
        0F11  D380 0000         X
        0F13  0F80
        0F14  26E6
        0F15  135F
        0F16  1325
003753                                  *
003754                                  * INITIALIZE LEVEL TABLE
003755                                  *
003756                                          CALL    ZV$F,ILV,MASK1,C16
        0F17  FBC0 0003
        0F19  D380 0000         X
        0F1B  0F80
        0F1C  26F6
        0F1D  135F
        0F1E  1325
003757                                  *
003758                                  * INITIALIZE COUNT TABLE
003759                                  *
003760  0F1F  8A80 135F                         INC     <MASK1
003761                                          CALL    ZV$F,INMB,MASK1,C16
        0F21  FBC0 0003
        0F23  D380 0000         X
        0F25  0F80
        0F26  26D6
        0F27  135F
        0F28  1325
003762                                  *
003763                                  * FILL PRIORITY LIST
003764                                  *
003765                                          CALL    ZV$F,PRIST,ALLONE,C16
        0F29  FBC0 0003
        0F2B  D380 0000         X
        0F2D  0F80
        0F2E  2706
        0F2F  1324
        0F30  1325
003766                                  *
003767                                  *
003768                                  * CREATE INTERRUPT SAVE AREAS
003769                                  *
003770  0F31  8700 135F                         CL      <MASK1
003771                                          CALL    ZV$F,IS1,MASK1,CNFILL FILL SDB WITH 0'S
        0F33  FBC0 0003
        0F35  D380 0000         X
        0F37  0F80
        0F38  2600
        0F39  135F
        0F3A  0F58
003772  0F3B  BB80 2600                         LAB     $B3,<IS1                START OF ISA'S
003773  0F3D  8751                              CL      =$R1                    R1 IS INDEX
003774                                  *
003775  0F3E  9CF3                      MISA    LDB     $B1,+$B3                BUMP PAST ISP
003776  0F3F  9873                              LDR     $R1,+$B3                BUMP PAST CHANNEL, DEVICE
003777  0F40  C870 7884                         LDR     $R4,=X'7884'            MASK
003778  0F42  CF73                              STR     $R4,+$B3                MASK
003779  0F43  C873                              LDR     $R4,+$B3                BUMP PAST SECOND MASK
003780  0F44  9B80 0F91                         LAB     $B1,<IHD1
003781  0F46  9FF3                              STB     $B1,+$B3                P
003782  0F47  4CFF                              LDV     $R4,=-1
003783  0F48  CF73                              STR     $R4,+$B3                S
003784                                  *
003785  0F49  9CF3                              LDB     $B1,+$B3                BUMP FOR B IN SAVE
003786  0F4A  BBC3 0005                         LAB     $B3,$B3,5
003787  0F4C  BD80 1344                         CMB     $B3,<ENDSAV
003788  0F4E  09F0                              BNE     >MISA                   BRANCH IF NOT DONE
003789                                  *
003790                                  *
003791                                  * CREATE LEVEL 0 SAVE AREA
003792                                  *
003793  0F4F  1CFF                      MISB    LDV     $R1,=-1
003794  0F50  9F00 26C2                         STR     $R1,<IOPSAV+1
003795  0F52  9F00 26C4                         STR     $R1,<IOPSAV+2+$AF
003796  0F54  8FC0 040F                         RSTR    SAV1,=Z'FFFF'           RESTORE ALL REG
        0F56  FFFF
003797  0F57  8381                              JMP     $B1
003798                                  *
003799  0F58  00D6                      CNFILL DC       (15/+5/*$AF)
003800                                  *
003801                                  * ROUTINES TO SUPPORT INTERRUPT SERVICING
003802                                  *
003803                                  *******
003804                                  *
003805                                  ***
003806                                  *
003807                                  *  FILL 16 WORD DATA TABLE,
003808                                  *  USED TO FILL ILV,IST,PRIST,NXRNG,CFIG,INMB
003809                                  *
003810                                  *  CALLED BY
003811                                  *
003812                                  *      LNJ  $B1,<FTBL
003813                                  *      DC  XX  VALUE
003814                                  *      DC  YY  CHANNELS
003815                                  *      DC  <ZZ  TABLE ADDRESS
003816                                  *
003817  0F59  8F40 040A                 FTBL   SAVE    SAV1,=Z'70B0'           R1,R2,R3,B2,B3,I
        0F5B  70B0
003818  0F5C  A871                              LDR     $R2,+$B1                PICK UP NUMBER
003819  0F5D  B871                              LDR     $R3,+$B1                PICK UP MASK
003820  0F5E  1CF0                              LDV     $R1,=-16                SET COUNTER
003821  0F5F  ACF1                              LDB     $B2,+$B1
003822  0F60  BBC2 0010                         LAB     $B3,$B2.16              ADDRESS OF END OF TABLE
003823  0F62  3882                      SLVLP   BGEZ    $R3,>ITM7               BRANCH IF BIT 0 = 0
003824  0F63  AF13                              STR     $R2,$B3.$R1
003825  0F64  3011                      ITM7    SCL     $R3,1                   ROTATE R3
003826  0F65  17FD                              BINC    $R1,>SLVLP
003827  0F66  8FC0 03FD                         RSTR    SAV1,=Z'70B0'
        0F68  70B0
003828  0F69  8381                              JMP     $B1                     DONE
```

```
003829                                  *
003830                                  * CHECK EXPECTED NUMBER OF INTERRUPTS ON EACH CHANNEL
003831                                  *
003832  0F6A 8F40 03F9                  CHLV    SAVE    SAV1,=Z'3000'
        0F6C 3000
003833  0F6D 8753                               CL      =$R3                    WILL TRACK CHANNEL NUMBER
003834  0F6E 2CF0                               LDV     $R2,=-16
003835  0F6F 89B0 26F6                  CHLV1   CMZ     <ILV,$R3                CHECK IF CHANNEL WAS TO BE TESTED
003836  0F71 0906                               BE      >CHLV2
003837  0F72 89B0 26D6                          CMZ     <INMB,$R3
003838  0F74 0903                               BE      >CHLV2
003839  0F75 F380 11CB                          LNJ     $B7,<ERROR              WRONG NUMBER OF INTERRUPTS ON CHANNEL
003840  0F77 8AD3                       CHLV2   INC     =$R3                    BUMP CHANNEL NUMBER
003841  0F78 27F7                               BINC    $R2,>CHLV1
003842  0F79 8FC0 03EA                          RSTR    SAV1,=Z'3000'
        0F7B 3000
003843  0F7C 8381                               JMP     $B1                     RETURN
003844                                  *
003845                                  * SET UP LCT FOR CHANNEL SPECIFIED IN R3
003846                                  *
003847                                  *               LNJ $B3,<SETLCT
003848                                  *                 -DC   TABLE
003849                                  *
003850  0F7D 8F40 03F6                  SETLCT  SAVE    SAV2,=Z'E8E0'           R1,<R2,<R4,<B2,<B1
        0F7F E8E0
003851  0F80 9CF3                               LDB     $B1,+$B3                GET ADDRESS OF TABLE
003852  0F81 8751                               CL      =$R1
003853                                  *
003854  0F82 A811                       LCT4    LDR     $R2,$B1,$R1             GET BYTE TO OUTPUT
003855  0F83 8AD1                               INC     =$R1
003856  0F84 2985                               BNEZ    $R2,>LCT5               BRANCH IF NOT AT END OF TABLE
003857  0F85 8FC0 03EE                          RSTR    SAV2,=Z'E8E0'
        0F87 E8E0
003858  0F88 8383                               JMP     $B3                     RETURN
003859                                  *
003860  0F89 C800 13C0                  LCT5    LDR     $R4,<CONT5              FUNCTION CODE FOROUT LCT BYTE
003861  0F8B C380 10EC                          LNJ     $B4,<CGSCH              FORM IO CONTROL WORD
003862  0F8D 8052                       LCT3    IO      =$R2,=$R4               OUTPUT BYTE
        0F8E 0054
003863  0F8F 07FE                               BIOF    >LCT3                   CHECK IF TAKEN
003864  0F90 0FF2                               B       >LCT4                   GET NEXT BYTE
003865                                  ********************
003866                                  *
003867                                  *  INTERRUPT HANDLER ROUTINE
003868                                  *
003869                                  ********************
003870                                  *
003871  0F91 8C51                       IHD1    STS     =$R1                    STORE STATUS IN R1
003872  0F92 9570 003F                          AND     $R1,=X'3F'              STRIP TO LEVEL
003873  0F94 B878 0000                          LDR     $R3,$IV,0               GET CHANNEL NUMB
003874  0F96 B570 FC00                          AND     $R3,=Z'FC00'            STRIP TO MLCP ADDRESS
003875  0F98 B900 133D                          CMR     $R3,<MLCADD
003876  0F9A 0905                               BE      >IHD05                  BRANCH IF GOOD
003877  0F9B 8E70 0080                          LEV     =X'0080'                QUICK CHANGE TO LEVEL 0
003878  0F9D F380 11CB                          LNJ     $B7,<ERROR              MLCP GAVE CP WRONG ADDRESS AFTER INTERRUPT
003879  0F9F B878 0000                  IHD05   LDR     $R3,$IV,0
003880  0FA1 B570 03C0                          AND     $R3,=X'3C0'             STRIP TO CHANNEL INFO
003881  0FA3 BF54                               STR     $R3,=$R4
003882  0FA4 3046                               SOR     $R3,6                   GETS SUBCHAN IN R3
003883  0FA5 89B0 26D6                          CMZ     <INMB,$R3               COMPARE NUMBER OF<INTERRUPTS
003884  0FA7 0985                               BNE     >IHD01
003885  0FA8 8E70 0080                          LEV     =X'0080'                SWITCH TO LEVEL 0
003886  0FAA F380 11CB                          LNJ     $B7,<ERROR              TOO MANY INTERRUPTS ON THIS CHANNEL
003887                                  *
003888  0FAC 0801 0005                  IHD01   BAL     IHD02                   BRANCH IF NEG
003889  0FAE 8E70 0080                          LEV     =X'0080'                SWITCH TO LEVEL 0
003890  0FB0 F380 11CB                          LNJ     $B7,<ERROR              ERROR, UNEXPECTED INTERRUPT
003891  0FB2 9930 26F6                  IHD02   CMR     $R1,<ILV,$R3            COMPARE LEVELS
003892  0FB4 0905                               BE      >IHD03
003893  0FB5 8E70 0080                          LEV     =X'0080'                SWITCH TO LEVEL 0
003894  0FB7 F380 11CB                          LNJ     $B7,<ERROR              WRONG CHANNEL INTERRUPTED THIS LEVEL
003895  0FB9 8980 134B                  IHD03   CMZ     <SCFLG                  CHECK SCAN FLAG
003896  0FBB 098E                               BNE     >IHD04                  IF SET, NO VALID CCB STATUS
003897  0FBC C600 13BF                          XOR     $R4,<CONT4              FORMS STATUS
003898  0FBE 8052                               IO      =$R2,=$R4               READ NEXT STATUS
        0FBF 0054
003899  0FC0 0F01 FFF8                          NOP     IHD03                   DELAY
003900  0FC2 A930 26E6                          CMR     $R2,<1ST,$R3
003901  0FC4 0905                               BE      >IHD04
003902  0FC5 8E70 0080                          LEV     =X'0080'                SWITCH TO LEVEL 0
003903  0FC7 F380 11CB                          LNJ     $B7,<ERROR              STATUS ERROR AFTER INTERRUPT
003904  0FC9 8E70 8004                  IHD04   LEV     =Z'8004'                SUSPEND, SCAN, AND DISPATCH LEV 4
003905  0FCB 0F81 FFC5                          B       IHD1
003906                                  ********************
003907                                  *
003908                                  * INTERRUPT MONITOR ROUTINE - LEVEL 4
003909                                  *
003910                                  ********************
003911  0FCD DC80 0EC8                  IMON    LDB     $B5,<STKPTR             GET PRIORITY STACK POINTER
003912  0FCF DD80 0EC7                          CMB     $B5,<PRIEND             CHECK IF AT END
003913  0FD1 0284                               BGE     >IMON2                  STACK IS FULL
003914                                  *
003915  0FD2 BF75                               STR     $R3,+$B5                STORE CHANNEL ON STACK
003916  0FD3 DF80 0EC8                          STB     $B5,<STKPTR             STORE NEW STACK POINTER
003917                                  *
003918  0FD5 8AB0 26D6                  IMON2   INC     <INMB,$R3               INCREMENT INTERRUPT COUNT
003919                                  *
003920  0FD7 8E70 803F                  IMON4   LEV     =Z'803F'                SCHEDULE LEV 63 AND SUSPEND
003921  0FD9 0FF4                               B       >IMON                   PICK UP NEXT TIME FROM HERE
003922                                  *
003923                                  * BLOCK WRITE DATA TO RAM
003924                                  *
003925                                  *       LNJ     $B1,<SDATA
003926                                  *       DC      DATA                    LOCATION OF DATA
003927                                  *       DC      RANGE                   NUMBER OF DATA BYTES
003928                                  *       DC      RAMAD                   RAM ADDRESS
003929                                  *       DC      EVEN                    0 = EVEN BYTE CPU ADDRESS
003930                                  *                                           BIT 15 = 1 FOR ODD BYTE START
003931                                  *                                           BIT 0 = 0 FOR 25 MS DELAY BEFORE RETURNIN
003932                                  *                                           BIT 0 = 1 FOR 250 MS DELAY BEFORE RETURNI
003933                                  *
003934                                  * R3 MUST CONTAIN THE CHANNEL NUMBER
003935                                  *
```

```
003936  0FDA  8F40 0399          SDATA  SAVE   SAV2,=Z'FFBF'        SAVE ALL BUT B1
        0FDC  FFBF
003937  0FDD  DCF1                       LDB    $B5,+$B1            GET ADDRESS OF DATA
003938  0FDE  C871                       LDR    $R4,+$B1            GET RANGE
003939  0FDF  DF80 0FEA                   STB    $B5,<SPRG5
003940  0FE1  A871                       LDR    $R2,+$B1            GET RAM ADDRESS
003941  0FE2  AF00 0FEB                   STR    $R2,<SPRG2
003942  0FE4  9871                       LDR    $R1,+$B1            LOAD START BYTE INDEX
003943  0FE5  9F57                       STR    $R1,=$R7            STORE  BYTE INDEX
003944  0FE6  9570 7FFF                  AND    $R1,=X'7FFF'
003945  0FE8  C380 1102                  LNJ    $B4,<MCCB           FORM CCB
003946  0FEA  0000             SPRG5  RESV   $AF,0                CPU ADDRESS
003947  0FEB  0000             SPRG2  DC     0                    RAM ADDRESS
003948  0FEC  C380 10F0                  LNJ    $B4,<CHCT           DO CHANNEL CONTROL
003949  0FEE  0F82                       B      >$+2
003950  0FEF  0400                       DC     Z'0400'             BLOCK WRITE
003951                         *
003952                         * PROGRAM ARIVES HERE FROM SDATA OR RDATA AS WELL AS SPRG.
003953                         *
003954  0FF0  7880 0FF7          SPRG3  BGEZ   $R7,<SPRG7
003955  0FF2  C380 1141                  LNJ    $B4,<TEST           WAIT FOR STATUS COMPLETE
003956  0FF4  0F82                       B      >$+2
003957  0FF5  001E                       DC     30                  250 MS TIMEOUT
003958  0FF6  0F85                       B      >SPRG8
003959  0FF7  C380 117D          SPRG7  LNJ    $B4,<DLAY           DELAY 25 MS
003960  0FF9  C380 10DA                  LNJ    $B4,<INXT           INPUT NEXT STATUS
003961  0FFB  82D5               SPRG8  LB     =$R5,=X'1000'       GET STATUS COMPLETE BIT
        0FFC  1000
003962  0FFD  0503                       BBT    >$+2+$AF
003963  0FFE  F380 11CB                  LNJ    $B7,<ERROR          STATUS COMPLETE NOT SET AFTER BLOCK WRITE
003964  1000  82D5                       LB     =$R5,=7
        1001  0007
003965  1002  0583                       BBF    >$+2+$AF
003966  1003  F380 11CB                  LNJ    $B7,<ERROR          ERROR, PARITY,MEMORY,OR RESOURCES
003967  1005  8F80 1374                  RSTR   <SAV2,=Z'FFBF'
        1007  FFBF
003968  1008  8381                       JMP    $B1
003969                         ******
003970                         *
003971                         * BLOCK READ FROM RAM.- CHAN. NUMBER MUST BE IN R3.
003972                         *
003973                         *        LNJ     $B1,<RDATA
003974                         *        DC      INBUFF              INPUT BUFFER ADDRESS
003975                         *        DC      RANGE               NUMBER OF BYTES
003976                         *        DC      RAMAD               RAM ADDRESS
003977                         *        DC      EVEN                0 = EVEN BYTE CPU ADDRESS
003978                         *                                        BIT 15 = 15 FOR ODD BYTE ADDRESS
003979                         *                                        BIT 0 = 1 FOR 250 MS DELAY AFTER STARTING
003980                         *                                        BIT 0 = 0 FOR 25 MS DELAY AFTER STARTING
003981                         *
003982  1009  8F40 036A          RDATA  SAVE   SAV2,=Z'FFBF'        SAVE EVERYTHING BUT B1.
        100B  FFBF
003983  100C  DCF1                       LDB    $B5,+$B1            GET IN BUFF ADD
003984  100D  DF80 1019                   STB    $B5,<RDTA1
003985  100F  C871                       LDR    $R4,+$B1            GET RANGE IN BYTES
003986  1010  9871                       LDR    $R1,+$B1
003987  1011  9F00 101A                   STR    $R1,<RDTA2
003988  1013  9871                       LDR    $R1,+$B1            PICK UP EVEN, ODD FLAG
003989  1014  9F57                       STR    $R1,=$R7            STORE BYTE  INDEX
003990  1015  9570 7FFF                  AND    $R1,=X'7FFF'
003991  1017  C380 1102                  LNJ    $B4,<MCCB           FORM CCB
003992  1019  1341               RDTA1  DC     <DUMMY               CPU ADDRESS
003993  101A  0000               RDTA2  DC     0                    RAM ADDRESS
003994  101B  C380 10F0                  LNJ    $B4,<CHCT           DO CHANNEL CONTROL
003995  101D  0F82                       B      >$+2
003996  101E  0800                       DC     Z'0800'             BLOCK READ
003997                         *
003998  101F  0F81 FFD0                   B      SPRG3               EXIT
003999                         *
004000                         * TURN ON ALL CHANNELS TO BE TESTED
004001                         *
004002                         *      LNJ $B3,<TRNON
004003                         *      DC CONT CHAN CONTROL WORD
004004                         *      DC XX 1= XMIT ONLY, 2= REC. ONLY, 0= BOTH
004005                         *
004006  1021  8F40 0342          TRNON  SAVE   SAV1,=Z'1F00'        R3,4,5,6
        1023  1F00
004007  1024  D873                       LDR    $R5,+$B3            GET CHANNEL CONTROL WORD
004008  1025  E873                       LDR    $R6,+$B3            GET TYPE FLAG
004009  1026  3C10                       LDV    $R3,=16             R3 GETS CHANNEL NUMBERS
004010  1027  3705               ITM11  BDEC   $R3,>TALL            BRANCH IF NOT DONE
004011  1028  8FC0 033B                   RSTR   SAV1,=Z'1F00'
        102A  1F00
004012  102B  8383                       JMP    $B3                 ALL DONE, EXIT
004013                         *
004014  102C  C800 13C2          TALL   LDR    $R4,<CONT7
004015  102E  89B0 26F6                   CMZ    <ILV,$R3            CHECK IF LEVEL SET UP
004016  1030  0977                       BE     >ITM11              BRANCH IF NOT SET UP
004017  1031  6904                       BEZ    $R6,>ITM11A         CHECK TYPE OF XFER
004018  1032  F856                       LDR    $R7,=$R6
004019  1033  F653                       XOR    $R7,=$R3
004020  1034  7BF3                       BODD   $R7,>ITM11          BRANCH IF NOT RIGHT TYPE
004021                         *
004022  1035  C380 10EC          ITM11A LNJ    $B4,<CGSCH          FORM IO CONTROL WORD
004023  1037  8055               ITM12  IO     =$R5,=$R4           START IO
        1038  0054
004024  1039  07FE                       BIOF   >ITM12              WAIT
004025  103A  0FEB                       B      >ITM11              DO NEXT
004026                         *
004027                         * SET LEVEL + CHANNEL FOR ALL ACTIVE CHANNELS
004028                         *
004029  103B  8753               SETV   CL     =$R3
004030  103C  9830 26F6          SETV2  LDR    $R1,<ILV,$R3        GET LEVEL
004031  103E  190A                       BEZ    $R1,>SETV1
004032  103F  C800 13C5                   LDR    $R4,<CONT11         GET FUNCTION CODE
004033  1041  C3C0 00AA                   LNJ    $B4,CGSCH           OR IN CHANNEL
004034  1043  8051                       IO     =$R1,=$R4
        1044  0054
004035  1045  0703                       BIOT   >$+2+$AF
004036  1046  F380 11CB                  LNJ    $B7,<ERROR          COMMAND NEVER SHOULD BE NAKED
004037                         *
004038  1048  8AD3               SETV1  INC    =$R3
004039  1049  3D10                       CMV    $R3,=16
```

```
004040  104A  09F2                    BNE   >SETV2
004041  104B  8383                    JMP   $B3
004042                          *
004043                          * SET UP INTERRUPT VECTORS FOR ENTRIES IN LEVEL TABLE
004044                          *
004045  104C  8752              SETVEC CL    =$R2
004046  104D  DB80 2601                LAB   $B5,<IS1+$AF          GET IS1 ISA SAVE POINTER
004047  104F  0F04              SET1   NOP   >$+4
004048  1050  BBC5 000C                LAB   $B3,$B5.9+3*$AF       FORM ADDRESS OF NEXT ISA
004049  1052  9820 26F6          SAF4   LDR   $R1,<ILV.$R2          GET LEVEL
004050  1054  89D1              SAF5   CMZ   =$R1
004051  1055  0903                     BE    >SET2                 BRANCH IF NO LEVEL SET FOR CHANNEL
004052  1056  0F90 0000       X        STB   $B5,<ZH1SAZ.$R1       STORE VECTOR
004053  1058  8AD2              SET2   INC   =$R2                  SET FOR NEXT CHANNEL
004054  1059  8FD5                     STB   $B3,=$B5
004055  105A  2D11                     CMV   $R2,=17               CHECK FOR END
004056  105B  0274                     BL    >SET1                 MORE
004057  105C  8384                     JMP   $B4                   EXIT
004058                          *
004059                          *
004060                          *          FIND FIRST ACTIVE CHANNEL
004061                          *
004062                          *            $R3-START CHECK ON THIS CHANNEL AND
004063                          *                CYCLE THROUGH ALL CHANNEL IF
004064                          *                ONE FOUND ON CHANNEL NUMBER IS HERE
004065                          *
004066                          *            LNJ  $B4,<FLN
004067                          *            <RETURN> IF NO LINES ON
004068                          *            <RETURN>
004069                          *
004070                          *
004071  105D  8F40 0306          FLN    SAVE  SAV1,=Z'A0A0'         SAVE REG. $R2,$B2
      105F  A0A0
004072  1060  2CFF                     LDV   $R2,=-1
004073  1061  AB80 1329                LAB   $B2,<AILI             ADDRESS OF ACTIVE LINE TABLE
004074  1063  3D10              FLN2   CMV   $R3,=16               SEE IF AT BOTTOM OF TABLE
004075  1064  0285                     BGE   >FLN3
004076  1065  A97E                     CMR   $R2,$B2.+$R3          SEE IF INDEX SCH. ON
004077  1066  097D                     BE    >FLN2
004078  1067  88D3                     DEC   =$R3                  SET $R3 BACK TO VALUE FOUND
004079  1068  A874                     LDR   $R2,+$B4              INCREMENT TO VALID RETURN
004080  1069  8FC0 02FA        FLN3   RSTR  SAV1,=Z'A0A0'         RESTORE REG.
      106B  A0A0
004081  106C  8384                     JMP   $B4                   JUMP OUT OF ROUTINE
004082                          *
004083                          * GENERAL INITIALIZE FOR MLCC
004084                          *
004085  106D  8F40 0316          GENITZ SAVE  SAV3,=Z'1808'         SAVE B4,R4
      106F  1808
004086  1070  8753                     CL    =$R3
004087  1071  C380 105D                LNJ   $B4,<FLN              FIND ACTIVE LINE
004088  1073  0000                     HLT                         NO ACTIVE LINES
004089  1074  C800 13C3                LDR   $R4,<CONT9            GET FUNCTION CODE FOR INITIALIZE
004090  1076  C380 10EC                LNJ   $B4,<CGSCH            MODIFY FOR CHANNEL
004091  1078  8070 8000                IO    =Z'8000',=$R4         INITIALIZE
      107A  0054
004092  107B  0703                     BIOT  >$+2+$AF
004093  107C  F380 11CB                LNJ   $B7,<ERROR            INITIALIZE WAS NAK'ED
004094  107E  C380 1184                LNJ   $B4,<DLAYLG           WAIT 225 MS
004095                          *
004096                          * TEST FOR BREAK
004097                          *
004098                                CALL  ZV$BRK
      1080  FBF0 0001
      1082  D380 0000       X
004099  1084  89C0 0000       P        CMZ   ZV$BRF
004100  1086  0980 01C9                BNE   <ROUTBG               BACK TO "NEXT"
004101  1088  8FC0 02FB                RSTR  SAV3,=Z'1808'
      108A  1808
004102  108B  8384                     JMP   $B4                   RETURN
004103                          *
004104                          *
004105                          *
004106                          *
004107                          *  PSEUDO - RANDOM DATA GENERATION ROUTINE
004108                          *           DATA IS BASED ON CRC GENERATOR
004109                          *
004110                          *      2 MODES OF OPERATION.
004111                          * *          INPUT - TAKES INITIAL CHAR AS BASE AND FORMS
004112                          *             TABLE OF DATA BASED ON CRC SPECIFIED.
004113                          *
004114                          *             OUTPUT - TAKES TABLE OF DATA AND FORMS ACCUMULATED CRC.
004115                          *EE    ONLY FOR CRC16 AND CCITT.
004116                          *
004117                          *      TO USE, MUST SET:
004118                          *             R6 - NUMBER OF BITS IN CHAR.
004119                          *             R7 - RANGE IN BYTES
004120                          *             B3 - TABLE ADDRESS
004121                          *             B6 - ADDRESS OF CRC TYPE.
004122                          *
004123                          *      FOR INPUT SET:
004124                          *
004125                          *             R5 - INITIAL CHAR. TO GIVE TO CRC GEN.
004126                          *
004127                          *      FOR OUTPUT SET:
004128                          *
004129                          *             R5 - MUST BE SET NEGATIVE.
004130                          *           CRC CHECK AND FILL ROUTINE
004131                          *
004132                          *
004133                          *           LNJ     $B4,<CRC
004134                          *
004135                          *
004136                          *      $B3 - ADDRESS WHERE CRC ACCUMULATOR IS STORED
004137                          *            OR CHARACTER IS LOADED IF $R5<0
004138                          *      $B6 - ADDRESS OF CRC TYPE
004139                          *
004140                          *
004141                          *      $R5 - IF >=0 CONTAINS INITIAL CHARACTER FOR FILLING TABLE
004142                          *            IF <0 NOT USED EXCEPT AS A FLAG FOR OUTPUT
004143                          *      $R6 - CONTAINS NUMBER OF BITS IN CHARACTER
004144                          *      $R7 - THE NUMBER OF CRC TO BE
004145                          *            GENERATED INTO/FROM THE TABLE AT $B3
```

```
004146                                  *
004147                                  *       THE ACCUMULATED CRC IS STORED IN LOCATION CRCAC.
004148                                  *
004149                                  *
004150  108C  8F40 0307                CRC     SAVE    SAV4,=Z'F994'
        108E  F994
004151  108F  C876                            LDR     $R4,+$B6                PUT CRC TYPE INTO $R4
004152  1090  8751                            CL      =$R1
004153  1091  8700 1348                       CL      <TEMP
004154  1093  8700 1349                       CL      <TEMP1
004155  1095  8753                            CL      =$R3                    CLEAR $R3 FOR LONG SHIFT
004156  1096  8752                            CL      =$R2
004157  1097  8256                            NEG     =$R6
004158  1098  EF00 1357                       STR     $R6,<IMGA+($AF-1)       STORE COUNTER
004159  109A  5885                            BGEZ    $R5,>CRCAA1             SEE IF OUTPUT OR INPUT
004160  109B  D0FF                            LDH     $R5,$B3.+$R3
004161  109C  8A80 1348                       INC     <TEMP
004162  109E  7001                            SOL     $R7,1
004163  109F  8257                CRCAA1      NEG     =$R7
004164  10A0  89D4                CRCAA       CMZ     =$R4                    IF LRC8
004165  10A1  0901 0035                       BE      CRCF                    BRANCH TO CRCF
004166  10A3  C970 F010                       CMR     $R4,=Z'F010'            IF CRC12
004167  10A5  091A                            BE      >CRCD                   BRANCH TO CRCD
004168  10A6  DF51                CRCA        STR     $R5,=$R1
004169  10A7  5041                            SOR     $R5,1
004170  10A8  9652                            XOR     $R1,=$R2
004171  10A9  2041                            SOR     $R2,1
004172  10AA  1B02                            BEVN    $R1,>CRCB
004173  10AB  A654                            XOR     $R2,=$R4
004174  10AC  67FA                CRCB        BINC    $R6,>CRCA
004175  10AD  89C0 029A                       CMZ     TEMP
004176  10AF  098E                            BNE     >CRCDD
004177  10B0  D852                            LDR     $R5,=$R2
004178  10B1  5048                            SOR     $R5,8
004179  10B2  88D5                            DEC     =$R5
004180  10B3  AF73                CRCC        STR     $R2,+$B3                STORE CRC ACCUM. IN TABLE
004181  10B4  E800 1357           CRCC2       LDR     $R6,<IMGA+($AF-1)       RESET CHAR. COUNT
004182  10B6  77EA                            BINC    $R7,>CRCAA              DO NEW CHARACTER
004183  10B7  AF00 1350                       STR     $R2,<CRCAC              STORE LAST CRC ACCUMULATED
004184  10B9  8FC0 02DA                       RSTR    SAV4,=Z'F994'           RESTORE REGS.
        10BB  F994
004185  10BC  8384                            JMP     $B4
004186                                  *
004187  10BD  D0FF                CRCDD       LDH     $R5,$B3.+$R3
004188  10BF  0FF6                            B       >CRCC2
004189                                  *
004190  10BF  5004                CRCD        SOL     $R5,4
004191  10C0  DF51                            STR     $R5,=$R1
004192  10C1  5045                            SOR     $R5,5
004193  10C2  9652                            XOR     $R1,=$R2
004194  10C3  2045                            SOR     $R2,5
004195  10C4  2004                            SOL     $R2,4
004196  10C5  82D1                            LB      =$R1,=X'10'
        10C6  0010
004197  10C7  0582                            BBF     >CRCE
004198  10C8  A654                            XOR     $R2,=$R4
004199  10C9  67F6                CRCE        BINC    $R6,>CRCD
004200  10CA  30CA                            DOR     $R3,10                  FORM GENERATED CRC
004201  10CB  3042                            SOR     $R3,2                   ACCUMULATOR
004202  10CC  2002                            SOL     $R2,2
004203  10CD  3088                            DOL     $R3,8
004204  10CE  AF73                            STR     $R2,+$B3
004205  10CF  D852                            LDR     $R5,=$R2
004206  10D0  30C8                            DOR     $R3,8
004207  10D1  3002                            SOL     $R3,2
004208  10D2  2042                            SOR     $R2,2
004209  10D3  308A                            DOL     $R3,10
004210  10D4  5048                            SOR     $R5,8
004211  10D5  88D5                            DEC     =$R5
004212  10D6  0FDE                            B       >CRCC2
004213                                  *
004214  10D7  A655                CRCF        XOR     $R2,=$R5
004215  10D8  8AD5                            INC     =$R5
004216  10D9  0FDA                            B       >CRCC
004217                                  *
004218                                  *
004219                                  * INPUT NEXT STATUS TO R5
004220                                  *
004221  10DA  8F00 13A2           INXT        SAVE    <SAV6,=Z'0008'          B4
        10DC  0008
004222  10DD  C800 13BF                       LDR     $R4,<CONT4              GET CONTROL WORD FOR INPUT NEXT STATUS
004223  10DF  C380 10EC                       LNJ     $B4,<CGSCH              MODIFY FOR CHANNEL
004224  10E1  8055                            IO      =$R5,=$R4               GET NEXT STATUS
        10E2  0054
004225  10E3  0703                            BIOT    >$+2+$AF
004226  10E4  F380 11CB                       LNJ     $B7,<ERROR              INPUT NEXT STATUS WAS NAK'ED
004227  10E6  DF00 13BA                       STR     $R5,<STAT               STORE LAST STATUS READ
004228  10E8  8F80 13A2                       RSTR    <SAV6,=Z'0008'
        10EA  0008
004229  10EB  8384                            JMP     $B4
004230                                  *
004231                                  *
004232                                  *
004233                                  *
004234                                  *
004235                                  *                   CHANGE CHANNEL
004236                                  *
004237                                  *       $R3 -CONTAINS CHANNEL WANTED
004238                                  *       $R4 -CONTAINS I/O CONTROL WORD TO BE CHANGED
004239                                  *
004240                                  *       LNJ     $B4,<CGSCH
004241                                  *
004242                                  *
004243  10EC  3006                CGSCH       SOL     $R3,6                   SHIFT TO CHANNEL POSITION
004244  10ED  C453                            OR      $R4,=$R3                OR CHANNEL NUMBER INTO CONTROLWORD
004245  10EE  3046                            SOR     $R3,6                   SHIFT TO NORMAL POSITION
004246  10EF  8384                            JMP     $B4                     GO BACK
004247                                  *
004248                                  * OUTPUT CHANNEL CONTROL
004249                                  *
004250                                  *   LNJ  $B4,CHCT
004251                                  *   B  >$+2         RETURN
004252                                  *   DC   XX         XX = CHANNEL CONTROL
```

```
004253                                  *
004254  10F0  8F00 1364                CHCT    SAVE    <SAV1,=Z'0C0D'      R4,R5,B5,B7,B4
        10F2  0C0D
004255  10F3  D874                            LDR     $R5,+$B4            DUMMY
004256  10F4  D674                            LDR     $R5,+$B4            GET CONTROL WORD
004257  10F5  C800 13C2                       LDR     $R4,<CONT7          FUN CODE FOR CCB CONTROL
004258  10F7  C380 10EC                       LNJ     $B4,<CGSCH          FORM IO CONTROL WORD
004259  10F9  8055                            IO      =$R5,=$R4           OUTPUT CCB CONTROL
        10FA  0054
004260  10FB  0703                            BIOT    >CH2
004261  10FC  F380 11CB                       LNJ     $B7,<ERROR          ERROR, IO WAS NAK'ED
004262                                  *
004263  10FE  8F80 1364                CH2     RSTR    <SAV1,=Z'0C0D'
        1100  0C0D
004264  1101  8384                            JMP     $B4
004265                                  *
004266                                  *
004267                                  *
004268                                  *
004269                                  *       CCB FORMATION
004270                                  *
004271                                  *         $R1 - CONTAINS INDEX OF CPU ADDRESS
004272                                  *         $R3 - CONTAINS CHANNEL WANTED
004273                                  *         $R4 - CONTAINS RANGE (NUMBER OF BYTES)
004274                                  *
004275                                  *         LNJ     $B4,<MCCB
004276                                  *         DC      CPU ADDRESS
004277                                  *         DC      RAM ADDRESS NUMBER OR CHANNEL CONTROL WORD.
004278                                  *
004279                                  *
004280  1102  8F40 0261                MCCB    SAVE    SAV1,=Z'FDF4'       SAVES $B1,$B3,$B2,$B5,$R7,$R5,$R4,R2,& $R1
        1104  FDF4
004281  1105  ACF4                            LDB     $B2,+$B4            LOAD $B2 WITH CPU ADDRESS
004282  1106  A874                            LDR     $R2,+$B4            PUT RAM ADDRESS IN $R2
004283  1107  DED4                            SWB     $B5,=$B4            ALLOW $B4 TO BE USE IN SUBR. CALL
004284  1108  D854                            LDR     $R5,=$R4
004285  1109  C800 13DC                       LDR     $R4,<CONT1          LOAD $R4 WITH I/O CONTROL WORD
004286  110B  C380 10EC                       LNJ     $B4,<CGSCH
004287  110D  8192                            IOLD    $B2,$R1,=$R4,=$R5   OUTPUT ADDRESS AND RANGE
        110E  0054
        110F  0055
004288  1110  0703                            BIOT    >$+2+$AF
004289  1111  F380 11CB                       LNJ     $B7,<ERROR          IOLD WAS NAK'ED
004290  1113  C800 13BE                       LDR     $R4,<CONT3          LOAD $R4 WITH I/O CONTROL WORD
004291  1115  C380 10EC                       LNJ     $B4,<CGSCH          PUT I/O CONTROL WORD IN $R4
004292  1117  8052                            IO      =$R2,=$R4           OUTPUT MLCC RAM ADDRESS
        1118  0054
004293  1119  0703                            BIOT    >$+2+$AF
004294  111A  F380 11CB                       LNJ     $B7,<ERROR          OUTPUT CONTROL FLAG WAS NAK'ED
004295  111C  CED5                            SWB     $B4,=$B5            SWAP FOR SUBR. RETURN
004296  111D  8FC0 0246                       RSTR    SAV1,=Z'FDF4'       RESTORE REGS.
        111F  FDF4
004297  1120  8384                            JMP     $B4
004298                                  *
004299                                  * INPUT LCT BYTE
004300                                  *
004301  1121  DED4                     INBYTE SWB     $B5,=$B4
004302  1122  C800 13C4                       LDR     $R4,<CONT10         GET FUNCTION CODE
004303  1124  C380 10EC                       LNJ     $B4,<CGSCH          PUT IN CHANNEL NUMBER
004304  1126  8055                            IO      =$R5,=$R4           INPUT
        1127  0054
004305  1128  0703                            BIOT    >$+2+$AF
004306  1129  F380 11CB                       LNJ     $B7,<ERROR          IO WAS NAK'ED
004307  112B  D570 FF00                       AND     $R5,=Z'FF00'
004308  112D  DED4                            SWB     $B5,=$B4
004309  112E  8384                            JMP     $B4
004310                                  *
004311                                  *
004312                                  *
004313                                  *       PACK RAM INSTRUCTIONS
004314                                  *
004315                                  *
004316                                  *       $R5 - RANGE IN WORDS OF BUFFER WHERE PACKED INSTRUCTIONS GO.
004317                                  *
004318                                  *         LNJ     $B4,<PKRIN
004319                                  *         DC      ADDRESS OF UNPACKED INSTRUCTION
004320                                  *         DC      ADDRESS OF PACKED INSTRUCTIONS
004321                                  *
004322                                  *
004323  112F  8F40 0234                PKRIN   SAVE    SAV1,=Z'CCD0'       SAVE $R1,$R4,$R5,$B3,$B1
        1131  CCD0
004324  1132  8255                            NEG     =$R5
004325  1133  BCF4                            LDB     $B3,+$B4            LOAD $B3 WITH ADDRESSS OF UNPACKED INSTS.
004326  1134  9CF4                            LDB     $B1,+$B4            LOAD $B4 WITH ADDRESS OF PACKED INSTS.
004327  1135  C873                     PKR1   LDR     $R4,+$B3            GET FIRST UNPACKED INST. IN $R4
004328  1136  4008                            SUL     $R4,8               SHIFT TO LEFT HALF OF WORD
004329  1137  9873                            LDR     $R1,+$B3            LOAD NEXT WORD
004330  1138  9570 00FF                       AND     $R1,=X'FF'          STRIP TO 8 BITS
004331  113A  C651                            XOR     $R4,=$R1            PUT WORD TOGETHER
004332  113B  CF71                            STR     $R4,+$B1            STORE WORD IN THE PACKED LOCATION
004333  113C  57F9                            BINC    $R5,>PKR1           BRANCH TO PACK ALL WORDS
004334  113D  8FC0 0226                       RSTR    SAV1,=Z'CCD0'       RESTORE REGS.
        113F  CCD0
004335  1140  8384                            JMP     $B4
004336                                  *
004337                                  * WAIT FOR STATUS COMPLETE
004338                                  *   LNJ  $B4,<TEST
004339                                  *   B    $+2     RETURN
004340                                  *   DC   XX      TIMEOUT IN 120TH'S SEC.
004341                                  *
004342                                  *
004343  1141  8F00 1364                TEST    SAVE    <SAV1,=Z'4909'      R1,4,7,B4,7
        1143  4909
004344  1144  DB80 0000          X             LAB     $B5,<ZHPFR          CLEAR B5
004345  1146  8751                            CL      =$R1
004346  1147  9F00 0000          X             STR     $R1,<ZHRTCI         ZERO OUT RTC RESET VALUE
004347  1149  1C01                            LDV     $R1,=1
004348  114A  9F00 0000          X             STR     $R1,<ZHRTCL         SET FOR RUPT LEVEL 1
004349  114C  9874                            LDR     $R1,+$B4            DUMMY TO INCREMENT B4
004350  114D  9874                            LDR     $R1,+$B4            PICK UP TIMEOUT VALUE
004351  114E  9F00 0000          X             STR     $R1,<ZHRTCC         SET REAL TIME CLOCK
004352  1150  9F00 13B8                       STR     $R1,<INTM
004353  1152  0004                            RTCN
```

```
004354  1153  C380 10DA              LNJ   $B4,<INXT          INPUT NEXT STATUS
004355  1155  82D5                   LB    =$R5,=Z'1000'      TEST FOR STATUS COMPLETE
        1156  1000
004356  1157  050E                   BBT   >TESTZ1            BRANCH IF COMPLETE
004357  1158  8980 0000        X     CMZ   <ZHRTCC            TEST RTC
004358  115A  090B                   BE    >TESTZ1            TIMEOUT
004359                               *
004360  115B  C800 13C1              LDR   $R4,<CONT6         FUNCTION CODE
004361  115D  C380 10EC              LNJ   $B4,<CGSCH
004362  115F  8055                   IO    =$R5,=$R4          INPPT STATUS
        1160  0054
004363  1161  0774                   BIOT  >TESTZ             BRANCH MEANS TRY AGAIN
004364  1162  F380 11CB              LNJ   $B7,<ERROR         INPUT STATUS WAS NAK'ED
004365  1164  0000                   HLT
004366  1165  0005              TESTZ1 RTCF                   TURN RTC OFF
004367  1166  9800 13B8              LDR   $R1,<INTM          GET INITIAL TIME
004368  1168  9200 0000        X     SUB   $R1,<ZHRTCC        GET ELAPSED
004369  116A  C380 1170              LNJ   $B4,<UPTM          UPDATE TIME
004370                               *
004371  116C  8F80 1364              RSTR  <SAV1,=Z'4909'
        116E  4909
004372  116F  8384                   JMP   $B4                EXIT
004373                               *
004374                               *
004375  1170  9A00 13B7        UPTM  ADD   $R1,<TTOT          ADD ELAPSED TO TAL
004376  1172  9F00 13B7              STR   $R1,<TTOT
004377  1174  1041                   SOR   $R1,1              DEVIDE TICKS BY 2
004378  1175  9970 003C              CMR   $R1,=60
004379  1177  0205                   BL    >UPTM1
004380  1178  8700 13B7              CL    <TTOT              CLEAR TOTAL
004381  117A  8A80 13B6              INC   <SEC               BUMP SECONDS COUNTER
004382  117C  8384              UPTM1 JMP   $B4
004383                               *
004384                               * TIME DELAY OF 25 MSEC (APPROX)
004385                               *
004386                               *     LNJ  $B4,<DLAY
004387                               *
004388  117D  8F00 1364              DLAY  SAVE  <SAV1,=Z'CE8C'
        117F  CE8C
004389  1180  9800 13BB              LDR   $R1,<HRTZ          GET FREQ
004390  1182  1044                   SOR   $R1,4              SET FOR CLOCK/16
004391  1183  0F87                   B     >DLAY2
004392                               *
004393                               * LONG DELAY (APPROX 225 MS)
004394                               *
004395  1184  8F40 01DF        DLAYLG SAVE   SAV1,=Z'CE8C'
        1186  CE8C
004396  1187  9800 13BB              LDR   $R1,<HRTZ          GET CLOCK
004397  1189  1041                   SOR   $R1,1              GET CLOCK/4
004398  118A  DB80 0000        X     DLAY2 LAB   $B5,<ZHPFR   CLEAR B5
004399  118C  0F60 0001        X           STB   $B5,<ZH1SAZ+$AF  ZERO OUT LEV1 INTERRUPT VECTOR
004400  118E  9F00 0000        X           STR   $R1,<ZHRTCC  SET RTC CURRENT VALUE
004401  1190  C380 1170              LNJ   $B4,<UPTM          UPDATE TIME
004402  1192  8751                   CL    =$R1
004403  1193  9F00 0000        X     STR   $R1,<ZHRTC1        RTC RESET VALUE
004404  1195  1C01                   LDV   $R1,=1
004405  1196  9F00 0000        X     STR   $R1,<ZHRTCL
004406  1198  C800 13C1              LDR   $R4,<CONT6         INPUT STATUS FUNCTION
004407  119A  C380 10EC              LNJ   $B4,<CGSCH         PUT IN CHAN NUMBER
004408                               *
004409  119C  1C0A                   LDV   $R1,=10
004410  119D  0004                   RTCN                     TURN ON RTC TIMER
004411  119E  1700 11A5        DLAY1 BDEC  $R1,<DLAY4
004412  11A0  1C0A                   LDV   $R1,=10            RESTORE COUNTER
004413  11A1  4E06                   ADV   $R4,=6
004414                               *
004415                               * GIVE INPUT LCI BYTE TO TRY TO INDUCE INTERFERENCE
004416  11A2  8055                   IO    =$R5,=$R4
        11A3  0054
004417  11A4  4EFA                   ADV   $R4,=-6            GET BACK INPUT STAT FC
004418  11A5  E800 13D5        DLAY4 LDR   $R6,<ERMG+1        GET TEST LABLE
004419  11A7  E970 5349              CMR   $R6,=A'SI'
004420  11A9  0985                   BNE   >DLAY3             BRANCH MEANS NOT TEST SI
004421                               * IS SOFT INITIALIZE TEST. BYPASS STATUS INPUTS.
004422  11AA  8AD1                   INC   =$R1
004423  11AB  8980 0000        X     CMZ   <ZHRTCC
004424  11AD  0F87                   B     >DLAY5
004425                               *
004426  11AE  8980 0000        X     DLAY3 CMZ   <ZHRTCC
004427  11B0  8055                   IO    =$R5,=$R4          INPUT STAT
        11B1  0054
004428                               * HANG HERE IF MLCC DOES NOT RESPOND WITH A SECOND HALF READ
004429  11B2  DF00 13BA              STR   $R5,<STAT          STORE LAST STATUS READ
004430  11B4  0A00 119E        DLAY5 BAG   <DLAY1             WAIT TILL DONE
004431  11B6  0005                   RTCF                     TURN OFF TIMER
004432  11B7  8FC0 01AC              RSTR  SAV1,=Z'CE8C'
        11B9  CE8C
004433  11BA  8384                   JMP   $B4                RETURN
004434                               *
004435                               *  SET UP RTC AND START IT.
004436                               *
004437                               *     LNJ  $B4,<RTC
004438                               *     DC   XX   XX = 120'S OF A SECOND DELAY DESIRED
004439                               *
004440  11BB  9874                   RTC   LDR   $R1,*$B4     GET DELAY DESIRED
004441  11BC  9F00 0000        X           STR   $R1,<ZHRTC1    STORE INITIAL VALUE
004442  11BE  9F00 0000        X           STR   $R1,<ZHRTCC    STORE CURRENT VALUE
004443  11C0  CF80 1347              STB   $B4,<TPR
004444  11C2  C380 1170              LNJ   $B4,<UPTM          UPDATE TIME
004445  11C4  CC80 1347              LDB   $B4,<TPR
004446  11C6  1C03                   LDV   $R1,=3             LEVEL 3
004447  11C7  9F00 0000        X     STR   $R1,<ZHRTCL        FOR RTC
004448  11C9  0004                   RTCN                     TURN ON RTC
004449  11CA  8384                   JMP   $B4                RETURN
004450                               *
004451                               *      ERROR PRINT ROUTINE
004452                               *
004453                               *ERLNJ  $B7,<ERROR
004454                               *
004455  11CB  8F40 01D8        ERROR SAVE  SAV5,=Z'FFFF'      SAVE ALL REGISTERS
        11CD  FFFF
004456  11CE  1CF0                   LDV   $R1,=-16
004457  11CF  8753                   CL    =$R3               CLEAR INDEX
```

```
004458  11D0  EBC7 FFFE                    LAB     $B6,$B7,-2*$AF      DECREMENT $B7
004459  11D2  EF80 11D9                    STB     $B6,<ERR1+4+$AF     STORE ERROR ADDRESS FOR ERROR CALL
004460                             ERR1    CALL    ZV$ER,$,ERMG
        11D4  FBC0 0003
        11D6  D380 0000       X
        11D8  0F80
        11D9  11D4
        11DA  13D4
004461  11DB  EB80 133B                    LAB     $B6,<CRLF           PUT CR-L INTO CALL
004462  11DD  CB80 133C                    LAB     $B4,<SPACE
004463  11DF  AB80 11F1                    LAB     $B2,<ERR3+4+2*$AF
004470  11E1  EF82                ER11    STB     $B6,$B2
004471  11E2  8980 13B9                    CMZ     <ERF                CHECK FLAG FOR SHORT REPORT
004472  11E4  0900 11FE                    BE      <ERR5
004473  11E6  BBB0 13A4            ERR2    LAB     $B3,<SAV5,$R3       FETCH AND STORE REGISTER
004474  11E8  9B80 11F0                    LAB     $B1,<ERR3+4+$AF
004480  11EA  BF81                ERR21   STB     $B3,$B1
004488                             ERR3    CALL    ZV$IHZ,$,$
        11EB  FBC0 0003
        11ED  D380 0000       X
        11EF  0F80
        11F0  11EB
        11F1  11EB
004493  11F2  8AD3                ERR34   INC     =$R3                BUMP INDEX
004494  11F3  CF82                        STB     $B4,$B2
004495  11F4  3D06                        CMV     $R3,=7*$AF+1        SEE IF CR-LF NEEDED
004496  11F5  0984                        BNE     >ERR4
004497  11F6  EB80 133B                    LAB     $B6,<CRLF
004498  11F8  EF82                        STB     $B6,$B2             PUT CR-LF INTO CALL
004499  11F9  1780 11E6            ERR4    BINC    $R1,<ERR2           DO NEXT REGISTER
004500  11FB  0000                        HLT                         HALT AFTER ERROR
004501  11FC  0F00 11F2                    NOP     <ERR34              PLACE FOR HALT PATCH
004502  11FE  8F80 13A4            ERR5    RSTR    <SAV5,=Z'FFFF'      RESTORE REGISTERS
        1200  FFFF
004503  1201  8387                        JMP     $B7                 RETURN
004514                             SAFF    NULL
004515                             *
004516                             *
004517                             *
004518                             *
004519                             *
004520                             *       -VM-VL-VC-VC -VI-V-VN-VS-VT-VR-VU-VC-VT-VI-VO-V-VN -VT-VE-VS-VT , -VC-VH-
004521                             *
004522                             *
004523                             *
004524                             * SOURCE =MLCS1C-CCP
004525                             *JUNE 6, 1978
004526                             *
004527                             *       RAM INSTRUCTION TEST PROGRAM GENERATION
004528                             *
004529                             *     THIS SECTION GETS LOADED AT HEX 200
004530                             *
004531                             *
004532                             *       ORG     X'200'
004533          1202               INSTR1 EQU    $
004534                             *       LOC     ST
004535          0200               ST EQU X'0200'
004536                             *       B       BT1                 B BT1
004539  1202    E002               *       WAIT    UNCONDITIONAL       BRANCH ERROR
004540                             *       LOC     BT1
004541          0203               BT1 EQU X'0203'
004542                             *       LD      =1                  LD =1
004543  1203    0190               *       C       =1                  C =1
004544  1204    0192               *       BET     BT6                 BRANCH IF TRUE
004547  1205    01E1               *       LOC     BT5
004548          0209               BT5 EQU X'0209'
004551                             *       WAIT    BRANCH              ON EQUAL TRUE OR 'LD' OR 'C' ERROR
004552  1206    0201               *       LOC     BT6
004555          020A               BT6 EQU X'020A'
004556                             *       BEF     BT5                 BRANCH IF FALSE
004557  1207    F105               *       LOC     RMOP
004559          020C               RMOP EQU X'020C'
004560                             *       C       24                  COMPARE WITH LCT 24 (0)
004561  1208    5218               *       BEF     BTC
004564  1209    F102               *       LOC     BT6
004565          0210               BT6 EQU X'0210'
004566                             *       WAIT    BRANCH              ON EQUAL FALSE ERROR
004567                             *       LOC     BTC
004570          0211               BTC EQU X'0211'
004571                             *       BET     BT5
004574  120A    01E1               *       DEC     R                   GOES FROM 1 TO 0
004575                             *       BZT     BT12                BRANCH IF ZERO TRUE
004576  120B    F705               *       LOC     BT11
004578          0216               BT11 EQU X'0216'
004579                             *       WAIT    BRANCH              ON ZERO TRUE ERROR
004580                             *       LOC     BT12
004581  120C    E202               BT12 EQU X'0217'
004584                             *       BZF     BT17                BRANCH IF ZERO FALSE
004585  120D    01F2               *       DEC     R                   GOES FROM 0 TO FF
004586                             *       BZF     BT18
004589  120E    0405               *       LOC     BT17
004590          021C               BT17 EQU X'021C'
004591                             *       WAIT    BRANCH              ON ZERO FALSE ERROR
004592                             *       LOC     BT18
004593          021D               BT18 EQU X'021D'
004594                             *       BZT     BT11
004595  120F    F202               *       BLCF    BT1D                BRANCH IF "L" FALSE
004596
004599
004600
004601
004604
004605
004606
004607
004608
004609
004610
004611  1210    01E2
004614
004615  1211    F8F3
```

```
004618                          *       LOC     BT5D
004619          0221            BT5D EQU X'0221'
004620                          *       WAIT    BRANCH          ON 'L' ERROR
004621  1212    0201
004622                          *       LOC     BT1D
004623          0222            BT1D EQU X'0222'
004624                          *       BLCT    BT5D            BRANCH IF "L" TRUE
004627  1213    E3FE
004628                          *       ST      23              STORE R (FF) TO LCT 23
004631  1214    5117
004632                          *       C       23              COMPARE WITH LCT 23
004635  1215    5217
004636                          *       BET     BT4F
004639  1216    E102
004640                          *       WAIT    ERROR           IN STORE INST.
004641                          *       LOC     BT4F
004642          022B            BT4F EQU X'022B'
004643                          *       AND     23              AND R (FF) WITH LCT 23 (FF)
004644  1217    0153
004647                          *       C       =X'FF'           C =X'FF'
004648  1218    1792
004651                          *       BEF     BT4D
004652  1219    FFF1
004655                          *       AND     24              AND R (FF) WITH LCT 24 (0)
004656  121A    0D53
004659                          *       C       24              COMPARE R(0) WITH LCT 24 (0)
004660  121B    1852
004663                          *       BEF     BT4D
004664  121C    18F1
004667                          *       AND     23              AND R (0) WITH LCT 23 (FF)
004668  121D    0753
004671                          *       C       24              COMPARE R (0), LCT 24 (0)
004672  121E    1752
004675                          *       BET     BTAC
004676  121F    18E1
004679                          *       LOC     BT4D
004680          023D            BT4D EQU X'023D'
004681                          *       WAIT    AND             INSTRUCTION ERROR
004682  1220    0201
004683                          *       LOC     BTAC
004684          023E            BTAC EQU X'023E'
004685                          *       AND     24              AND R (0) WITH LCT 24 (0)
004688  1221    5318
004689                          *       C       24
004692  1222    5218
004693                          *       BEF     BT4D
004696  1223    F1FA
004697                          *       OR      24              OR R (0), LCT 24 (0)
004700  1224    5418
004701                          *       C       24              COMPARE R WITH DATA BUFF
004704  1225    5218
004705                          *       BEF     BT4C
004708  1226    F10D
004709                          *       OR      23              OR R(0), LCT 23 (FF)
004712  1227    5417
004713                          *       C       23              COMPARE R WITH LCT 23
004716  1228    5217
004717                          *       BEF     BT4C
004720  1229    F107
004721                          *       OR      24              OR R (FF), LCT 24 (0)
004724  122A    5418
004725                          *       C       23              COMPARE R (FF), LCT 23 (FF)
004728  122B    5217
004729                          *       BET     BTRC
004732  122C    E102
004733                          *       LOC     BT4C
004734          0256            BT4C EQU X'0256'
004735                          *       WAIT    OR              INSTRUCTION ERROR
004736                          *       LOC     BTRC
004737          0257            BTRC EQU X'0257'
004738                          *       OR      23              OR R (FF), LCT23 (FF)
004739  122D    0154
004742                          *       C       23              COMPARE R (FF), LCT 23
004743  122E    1752
004746                          *       BEF     BT4C
004747  122F    17F1
004750                          *       XOR     24              EXCL. OR R (FF), DATA BUFF  (0)
004751  1230    FA55
004754                          *       C       23              COMPARE R(FF), LCT 23 (FF)
004755  1231    1852
004758                          *       BEF     BT4B
004759  1232    17F1
004762                          *       XOR     23              EXCLUSIVE OR R (FF), LCT 23 (FF)
004763  1233    0D55
004766                          *       C       24              COMPARE R (0) WITH LCT 24 (0)
004767  1234    1752
004770                          *       BEF     BT4B
004771  1235    18F1
004774                          *       XOR     24              EXCL. OR R (0) WITH LCT 24 (0)
004775  1236    0755
004778                          *       C       24
004779  1237    1852
004782                          *       BET     BTXC
004783  1238    18E1
004786                          *       LOC     BT4B
004787          026F            BT4B EQU X'026F'
004788                          *       WAIT    XOR             INSTRUCTION ERROR
004789  1239    0201
004790                          *       LOC     BTXC
004791          0270            BTXC EQU X'0270'
004792                          *       XOR     23              EXCL. OR R (0), LCT 23 (FF)
004795  123A    5517
004796                          *       C       23              COMPARE R (FF), LCT 23  (FF)
004799  123B    5217
004800                          *       BEF     BT4B
004803  123C    F1FA
004804                          *       LOC     BT4J
004805          0276            BT4J EQU X'0276'
004806                          *       LD      =X'AA'           LD =X'AA'
004809  123D    90AA
004810
004811                          *       RET
004812  123E    0600            *       NOP
```

```
004813                          *
004814                          * RAM INSTRUCTION TEST, PART 2, LOADED AT 27A
004815                          *
004816                          *
004817          123F            INSTR2 EQU    $
004818  123F    E60B                   DC     Z'E60B'              B DFD
004819  1240    8101                   DC     Z'8101'
004820                          *      ORG    X'27E'
004821                          *      NOP    DUMMY
004822                          *      LOC    BTYD
004823          027F            BTYD EQU X'027F'
004824                          *      BS     S1                   BRANCH STORE TO X'200'
004825  1241    00F0
004828                          *      C      =X'AA'               COMPARE R(AA), =AA
004829  1242    8092
004832                          *      BET    BTA7
004833  1243    AAE1
004836                          *      LOC    BTB7
004837          0285            BTB7 EQU X'0285'
004838                          *      WAIT   BRANCH               AND STORE ERROR
004839  1244    0201
004840                          *      LOC    BTA7
004841          0286            BTA7 EQU X'0286'
004842                          *      BS     BTA6                 BRANCH STORE TO X'306'
004845  1245    F07F
004848                          *      C      =X'55'                C =X'55'
004849  1246    9255
004850                          *      BEF    BTB7
004853  1247    F1FA
004854                          *
004855                          * HERE FOLLOWS THE TABLE LOOKUP TEST
004856                          *
004857                          *      LD     =0                   LD =0
004860  1248    9000
004861                          *      TLU    55                   TABLE LOOKUP, POINTER AT LCT 55
004864  1249    5637
004865                          *      C      =3                   FIRST ENTRY OF TABLE IS 3
004868  124A    9203
004869                          *      BET    NEXT
004872  124B    E102
004873                          *      WAIT   TABLE                LOOKUP ERROR 24 1ST ENTRY
004874                          *      LOC    NEXT
004875          0295            NEXT EQU X'0295'
004876                          *      LD     =2                   LD =2, GET 3D ENTRY
004877  124C    0190
004880                          *      TLU    55                   TABLE LOOKUP, LCT 55
004881  124D    0256
004884                          *      WAIT
004885  124E    3701
004886                          *      WAIT
004887                          *      WAIT
004888  124F    0101
004889                          *      WAIT   TLU                  ERROR 24 INCORRECT BRANCH
004890                          *      B      NEXT1                GOOD
004891  1250    01E0
004894                          *      WAIT   BAD                  TLU BRANCH
004895  1251    0401
004896                          *      WAIT   DITTO
004897                          *      WAIT   DITTO
004898  1252    0101
004899                          *      LOC    NEXT1
004900          02A2            NEXT1 EQU X'02A2'
004901                          *      C      =X'2'                 COMPARE R WITH =X'2'
004904  1253    9202
004905                          *      BET    NEXT2
004908  1254    E102
004909                          *      WAIT   TLU                  ERROR 24 WRONG DATA CAME BACK
004910                          *      LOC    NEXT2
004911          02A7            NEXT2 EQU X'02A7'
004912                          *      LD     =1                   GET SECOND ENTRY FROM TABLE
004913  1255    0190
004916                          *      TLU    55                   TLU INDIRECT THRU LCT 55
004917  1256    0156
004920                          *      C      =X'1A'
004921  1257    3792
004924                          *      BET    NEXT3
004925  1258    1AE1
004928                          *      WAIT   WRONG                DATA BACK FROM TLU
004929  1259    0201
004930                          *      LOC    NEXT3
004931          02B0            NEXT3 EQU X'02B0'
004932                          *      WAIT
004933                          *      NOP
004934  125A    0100
004935                          *
004936                          * INSTRUCTION TEST, PART 3, GETS LOADED AT 306
004937                          *
004938                          *
004939                          *      ORG    X'306'
004940          125B            INSTR3 EQU    $
004941                          *      LOC    BTA6
004942          0306            BTA6 EQU X'0306'
004943                          *      LD     3                    LD  R FROM LCT 3 (FF), PRELOADED IN LCT SETU
004946  125B    5003
004947                          *      C      =X'FF'                C ='FF'
004950  125C    92FF
004951                          *      BEF    BTA9
004954  125D    F107
004955                          *      LD     21                   LD R, LCT 21 (AA), PRELOADED
004958  125E    5015
004959                          *      C      =X'AA'
004962  125F    92AA
004963                          *      BET    BTB5
004966  1260    E102
004967                          *      LOC    BTA9
004968          0312            BTA9 EQU X'0312'
004969                          *      WAIT   LOAD                 ADDRESSING ERROR
004970                          *      LOC    BTB5
004971          0313            BTB5 EQU X'0313'
004972                          *      LD     44                   LOAD R,LCT 44 (55), PRE LOADED
004973  1261    0150
004976                          *      C      =X'55'                =X'55'
004977  1262    2C92
```

```
004980                          *       BEF     BTA9
004981  1263  55F1
004984                          *       LD      63                      LOAD  R, LCT 63 (AA), PRELOADED
004985  1264  FA50
004988                          *       C       =X'AA'                   C =X'AA'
004989  1265  3F92
004992                          *       BEF     BTA9
004993  1266  AAF1
004996                          *       LD      =X'80'                  TEST VALUE
004997  1267  F490
005000                          *       SR      SHIFT                   RIGHT
005001  1268  8007
005002                          *       C       =X'40'
005005  1269  9240
005006                          *       BEF     ERR6                    SHIFT FAILED IF BRANCH
005009  126A  F10E
005010                          *       SR      SHIFT                   AGAIN
005011                          *       C       =X'20'
005012  126B  0792
005015                          *       BEF     ERR6                    BRANCH IF SHIFT FAILED
005016  126C  20F1
005019                          *       BLBF    BT4G                    BRANCH IF LAST BLOCK FALSE
005020  126D  09F4
005023                          *       LOC     BT4H
005024        032D              BT4H EQU X'032D'
005025                          *       WAIT    BLCF                    OR BLCT FAILURE
005026  126E  0201
005027                          *       LOC     BT4G
005028        032E              BT4G EQU X'032E'
005029                          *       BLBT    BT4H                    BRANCH IF LAST BLOCK TRUE
005032  126F  E4FE
005033                          *       LD      =X'55'                   LD =X'55'
005036  1270  9055
005037                          *       RET
005038                          *       LOC     ERR6
005039        0333              ERR6 EQU X'0333'
005040                          *       WAIT
005041  1271  0601
005042                          *       NOP
005043                          *
005044                          *
005045                          * INSTRUCTION TEST - PART 4 - LOADED AT X'5FD'
005046                          *
005047        1272              INSTR4 EQU      $
005048  1272  0101                      DC      Z'0101'                 WAIT,WAIT
005049  1273  E604                      DC      Z'E604'                  B X'B00'
005050  1274  FF00                      DC      Z'FF00'
005051                          *
005052                          * PART 5 - LOADED AT AFE
005053                          *
005054        1275              INSTR5 EQU      $
005055  1275  0101                      DC      Z'0101'                 WAIT,WAIT
005056  1276  E617                      DC      Z'E617'                 B 27E
005057  1277  7C01                      DC      X'7C01'
005058                          *
005059                          * INSTRUCTION TEST - PART 6, LOADED AT DFA
005060                          *
005061        1278              INSTR6 EQU      $
005062  1278  0101                      DC      Z'0101'                 WAIT,WAIT
005063  1279  E6F8                      DC      Z'E6F8'                  B X'5FF'
005064  127A  0001                      DC      Z'0001'
005065                          *
005066                          * TRANSMIT DATA CHANNEL PROGRAM
005067                          *               GETS LOADED AT X'400' IN MLCC RAM
005068                          *
005069                          *
005070                          *
005071                          *       ORG     X'400'
005072  127B  0000
005073                          *       LOC     XMDT
005074        0400              XMDT EQU X'0400'
005075        127C              TRANM1 EQU      $
005076                          *       B       RTD1
005079  127C  E002
005080                          *       LOC     RTD
005081        0402              RTD EQU X'0402'
005082                          *       WAIT    NORMAL                  END
005083                          *       LOC     RTD1
005084        0403              RTD1 EQU X'0403'
005085                          *       LD      55                      LOAD AND CHECK COUNTER
005086  127D  0150
005089                          *       C       =5                      IF 5 STORE IN 57
005090  127E  3792
005093                          *       BET     R57
005094  127F  05E1
005097                          *       C       =4                      IF 4 STORE IN 58
005098  1280  2C92
005101                          *       BET     R58
005102  1281  04E1
005105                          *       C       =3                      IF 3 STORE, IN 59
005106  1282  2D92
005109                          *       BET     R59
005110  1283  03E1
005113                          *       C       =2                      IF 2 STORE IN 60
005114  1284  2E92
005117                          *       BET     R60
005118  1285  02E1
005121                          *       C       =1                      IF 1 STORE IN 61
005122  1286  2F92
005125                          *       BET     R61
005126  1287  01E1
005129                          *       LD,     IS                      ZERO
005130  1288  3010
005131                          *       ST      62                      STORE IN 62
005134  1289  513E
005135                          *       LOC     RTD2
005136        041C              RTD2 EQU X'041C'
005137                          *       LD      56                      CHECK FLAG
005140  128A  5038
005141                          *       C       =0
005144  128B  9200
005145                          *       BET     R70                     BRANCH IF EOR
005148  128C  E109
```

```
005149                          *       BLCF    R67
005152  128D  F302
005153                          *       WAIT    GNB             BLCF DID NOT BRANCH
005154                          *       LOC     R67
005155          0425            R67 EQU X'0425'
005156                          *       BLCT    R68
005157  128E  01E3
005160                          *       GNB
005161  128F  0C02
005162                          *       B       RTD
005165  1290  E0D9
005166                          *
005167                          *
005168                          *
005169                          *       LOC     R70
005170          042A            R70 EQU X'042A'
005171                          *       BLCF    R69
005174  1291  F308
005175                          *       BLCT    R71
005178  1292  E302
005179                          *       WAIT    WAIT            BLCF DID NOT BRANCH
005180                          *       LOC     R71
005181          042F            R71 EQU X'042F'
005182                          *       GNB
005183  1293  0102
005184                          *       B       RTD
005187  1294  E0D1
005188                          *
005189                          *
005190                          *
005191                          *       LOC     R68
005192          0432            R68 EQU X'0432'
005193                          *       WAIT    BLCT            DID BRANCH
005194                          *       LOC     R69
005195          0433            R69 EQU X'0433'
005196                          *       WAIT    BLCF            DID BRANCH
005197  1295  0101
005198                          *
005199                          *
005200                          *
005201                          *       LOC     R57
005202          0434            R57 EQU X'0434'
005203                          *       LD
005204                          *       ST      57
005205  1296  1051
005206                          *       B       R56
005209  1297  39E0
005212                          *       LOC     R58
005213          0439            R58 EQU X'0439'
005214                          *       LD
005215  1298  1310
005216                          *       ST      58
005219  1299  513A
005220                          *       B       R56
005223  129A  E00E
005224                          *       LOC     R59
005225          043E            R59 EQU X'043E'
005226                          *       LD
005227                          *       ST      59
005228  129B  1051
005231                          *       B       R56
005232  129C  3BE0
005235                          *       LOC     R60
005236          0443            R60 EQU X'0443'
005237                          *       LD
005238  129D  0910
005239                          *       ST      60
005242  129E  513C
005243                          *       B       R56
005246  129F  E004
005247                          *       LOC     R61
005248          0448            R61 EQU X'0448'
005249                          *       LD
005250                          *       ST      61
005251  12A0  1051
005254                          *       LOC     R56
005255          044B            R56 EQU X'044B'
005256                          *       LD      55
005257  12A1  3D50
005260                          *       DEC
005261  12A2  3705
005262                          *       ST      55
005265  12A3  5137
005266                          *       LD      63
005269  12A4  503F
005270                          *       DEC
005271                          *       ST      63
005272  12A5  0551
005275                          *       BZT     RTD2
005276  12A6  3FE2
005279                          *       BLCF    RTD1
005280  12A7  C6F3
005283                          * COME HERE FOR UNDERRUN TEST ONLY
005284                          *       GNB
005285  12A8  AB02
005286                          *       LD
005287                          *       B       RTD
005288  12A9  10E0
005291                          *       NOP
005292  12AA  A600
005293                          *       NOP
005294                          *       NOP
005295  12AB  0000
005296                          *
005297                          *
005298                          *
005299                          * RECEIVE DATA CHANNEL PROGRAM
005300                          *
005301                          *
005302                          *
005303                          *       ORG     X'200'
005304          12AC            RCDTC1 EQU      *
005305                          *       LOC     RCDT
```

```
005306          0200    RCDT EQU X'0200'
005307                  *       B       XMD1
005310  12AC    E002
005311                  *       LOC     XMD
005312          0202    XMD EQU X'0202'
005313                  *       WAIT    NORMAL          END
005314                  *       LOC     XMD1
005315          0203    XMD1 EQU X'0203'
005316                  *       LD      55              LOAD AND CHECK COUNTER
005317  12AD    0155
005320                  *       C       =5              IF 5 LOAD FROM 57
005321  12AE    3792
005324                  *       BET     XM57
005325  12AF    05E1
005326                  *       C       =4              IF 4 LOAD FROM 58
005327  12B0    2992
005332                  *       BET     XM58
005333  12B1    04E1
005336                  *       C       =3              IF 3 LOAD FROM 59
005337  12B2    2A92
005340                  *       BET     XM59
005341  12B3    03E1
005344                  *       C       =2              IF 2 LOAD FROM 60
005345  12B4    2B92
005348                  *       BET     XM60
005349  12B5    02E1
005352                  *       C       =1              IF 1 LOAD FROM 61
005353  12B6    2C92
005356                  *       BET     XM61
005357  12B7    01E1
005360                  *       LD      62              IS ZERO
005361  12B8    2D50
005364                  *       ST,     PDI             IN DATA BUFFER
005365  12B9    3E11
005366                  *       LOC     XMD2
005367          021C    XMD2 EQU X'021C'
005368                  *       LD      56              CHECK FLAG
005371  12BA    5038
005372                  *       C       =0
005375  12BB    9200
005376                  *       BET     XM70
005379  12BC    E109
005380                  *       BLCF    XM67
005383  12BD    F302
005384                  *       WAIT    GNB             BLCF DID NOT BRANCH
005385                  *       LOC     XM67
005386          0225    XM67 EQU X'0225'
005387                  *       BLCT    XM68
005388  12BE    01E3
005391                  *       LOC     XM71
005392          0227    XM71 EQU X'0227'
005393                  *       GNB
005394  12BF    0902
005395                  *       B       XMD
005398  12C0    E0D9
005399                  *
005400                  *
005401                  *
005402                  *       LOC     XM70
005403          022A    XM70 EQU X'022A'
005404                  *       BLCF    XM69
005407  12C1    F305
005408                  *       BLCT    XM71
005411  12C2    E3FA
005412                  *       WAIT    WAIT            BLCT DID NOT BRANCH
005413                  *
005414                  *
005415                  *
005416                  *       LOC     XM68
005417          022F    XM68 EQU X'022F'
005418                  *       WAIT    BLCT            DID BRANCH
005419  12C3    0101
005420                  *       LOC     XM69
005421          0230    XM69 EQU X'0230'
005422                  *       WAIT    BLCF            DID BRANCH
005423                  *
005424                  *
005425                  *
005426                  *       LOC     XM57
005427          0231    XM57 EQU X'0231'
005428                  *       LD      57
005429  12C4    0150
005432                  *       ST
005433  12C5    3911
005434                  *       B       XM56
005437  12C6    E013
005438                  *       LOC     XM58
005439          0236    XM58 EQU X'0236'
005440                  *       LD      58
005443  12C7    503A
005444                  *       ST
005445                  *       B       XM56
005446  12C8    11E0
005449                  *       LOC     XM59
005450          023B    XM59 EQU X'023B'
005451                  *       LD      59
005452  12C9    0E50
005455                  *       ST
005456  12CA    3B11
005457                  *       B       XM56
005460  12CB    E009
005461                  *       LOC     XM60
005462          0240    XM60 EQU X'0240'
005463                  *       LD      60
005466  12CC    503C
005467                  *       ST
005468                  *       B       XM56
005469  12CD    11E0
005472                  *       LOC     XM61
005473          0245    XM61 EQU X'0245'
005474                  *       LD      61
005475  12CE    0450
005478                  *       ST
```

```
005479  12CF  3D11
005480                        *       LOC     XM56
005481        0248            XM56 EQU X'0248'
005482                        *       LD      55
005485  12D0  5037
005486                        *       DEC
005487                        *       ST      55
005488  12D1  0551
005491                        *       LD      63
005492  12D2  3750
005495                        *       DEC
005496  12D3  3F05
005497                        *       ST      63
005500  12D4  513F
005501                        *       BZT     XMD2
005504  12D5  E2C9
005505                        *       BLCF    XMD1
005508  12D6  F3AE
005509                        *       GNB     FOR                     CCB OVERRUN TEST ONLY
005510                        *       ST
005511  12D7  0211
005512                        *       B       XMD
005515  12D8  E0A9
005516                        *       NOP
005517                        *       NOP
005518  12D9  0000
005519                        *       NOP
005520                        *
005521                        *
005522                        *
005523                        *
005524                        *       MLCC INPUT DATA PROGRAM WITH
005525                        *         CRC TO GENERATE RANDOM DATA
005526                        *
005527                        *
005528                        *
005529                        *       ORG     X'200'
005530  12DA  0000
005531        12DB            RCRT1   EQU     $
005532                        *       LOC     RMCRT
005533        0200            RMCRT EQU X'0200'
005534                        *       LD      2                       CHECK CONF. IS LRC8
005537  12DB  5002
005538                        *       C       =X'C6'
005541  12DC  92C6
005542                        *       BEF     RMCRT3                  BRANCH IF NOT LRC8
005545  12DD  F111
005546                        *       LD      =1
005549  12DE  9001
005550                        *       LOC     RMCRT5
005551        0208            RMCRT5 EQU X'0208'
005552                        *       ST      24
005555  12DF  5118
005556                        *       CCH
005557                        *       BS      RMCRT4
005558  12E0  04F0
005561                        *       LD      24
005562  12E1  2250
005565                        *       XOR     =X'FF'
005566  12E2  1895
005569                        *       DEC
005570  12E3  FF05
005571                        *       XOR     =X'FF'
005574  12E4  95FF
005575                        *       B       RMCRT5
005578  12E5  E0F3
005579                        *       LOC     RMCRT3
005580        0216            RMCRT3 EQU X'0216'
005581                        *       C       =X'44'                  SEE IF CRC12
005584  12E6  9244
005585                        *       LD      =5
005586  12E7  9005
005589                        *       BET     RMCRT7
005592  12E8  E109
005593                        *       LOC     RMCRT1
005594        021C            RMCRT1 EQU X'021C'
005595                        *       CCH
005596                        *       BS      RMCRT4
005597  12E9  04F0
005600                        *       LD      4
005601  12EA  1050
005604                        *       DEC
005605  12EB  0405
005606                        *       B       RMCRT1
005609  12EC  E0F9
005610                        *
005611                        *       LOC     RMCRT7
005612        0224            RMCRT7 EQU X'0224'
005613                        *       CCH
005614                        *       BS      RMCRT4
005615  12ED  04F0
005618                        *       LD      4
005619  12EE  0850
005622                        *       DEC
005623  12EF  0405
005624                        *       AND     =X'3F'
005627  12F0  933F
005628                        *       B       RMCRT7
005631  12F1  E0F7
005632                        *       LOC     RMCRT4
005633        022E            RMCRT4 EQU X'022E'
005634                        *       LD      4
005637  12F2  5004
005638                        *       ST
005639                        *       LD      3
005640  12F3  1150
005643                        *       ST
005644  12F4  0311
005645                        *       BLCF    RMCRT8
005648  12F5  F302
005649                        *       GNB
005650                        *       LOC     RMCRT8
005651        0237            RMCRT8 EQU X'0237'
```

```
005652                          *       LD      23
005653  12F6  0250
005656                          *       DEC
005657  12F7  1705
005658                          *       ST      23
005661  12F8  5117
005662                          *       BZT     RMCRT2
005665  12F9  E202
005666                          *       RET
005667                          *       LOC     RMCRT2
005668          023F            RMCRT2 EQU X'023F'
005669                          *       WAIT
005670  12FA  0601
005671                          *       B       RMCRT
005674  12FB  E0BF
005675                          *       NOP
005676                          *
005677                          *
005678                          *        MLCC OUTPUT DATA PROGRAM
005679                          *
005680                          *
005681                          *
005682                          *       ORG     X'200'
005683  12FC  0000
005684          12FD            ROD1    EQU     $
005685                          *       LOC     RMOPT
005686          0200            RMOPT EQU X'0200'
005687                          *       LD      =4
005690  12FD  9004
005691                          *       ST      23
005694  12FE  5117
005695                          *       LOC     RMOPTA
005696          0204            RMOPTA EQU X'0204'
005697                          *       LD      =125
005700  12FF  907D
005701                          *       ST      24
005704  1300  5118
005705                          *       LOC     RMOPT1
005706          0208            RMOPT1 EQU X'0208'
005707                          *       LD
005708                          *       CCH
005709  1301  1004
005710                          *       LD
005711                          *       BLCF    RMOPT2                  CHECK IF LAST CHARACTER
005712  1302  10F3
005715                          *       GNB
005716  1303  0202
005717                          *       LOC     RMOPT2
005718          020E            RMOPT2 EQU X'020E'
005719                          *       CCH
005720                          *       LD      24
005721  1304  0450
005724                          *       DEC
005725  1305  1805
005726                          *       C       =0
005729  1306  9200
005730                          *       ST      24
005733  1307  5118
005734                          *       BEF     RMOPT1
005737  1308  F1F1
005738                          *       LD      23
005741  1309  5017
005742                          *       DEC
005743                          *       C       =0
005744  130A  0592
005747                          *       ST      23
005748  130B  0051
005751                          *       BEF     RMOPTA
005752  130C  17F1
005755                          *       WAIT
005756  130D  E401
005757                          *       NOP
005758                          *
005759                          *
005760                          *       NOP
005761  130E  0000
005762                          *
005763                          * BVT, BVF TEST
005764          130F            VALTST1 EQU     $
005765                          *       ORG     X'200'
005766                          *       LD      =4                      COUNTER FOR 4 CCB'S
005769  130F  9004
005770                          *       LOC     BVT
005771          0202            BVT EQU X'0202'
005772                          *       DATA    X'E7'                   BVT
005775                          *       DATA    X'3'                    +3
005778  1310  E703
005779                          *       WAIT    ERROR
005780                          *       WAIT    ERROR, BVT              SHOULD BRANCH
005781  1311  0101
005782                          *       DATA    X'F7'                   BVF
005785                          *       DATA    X'3'                    +3
005788  1312  F703
005789                          *       DATA    X'E0'                   B
005792                          *       DATA    X'3'                    +3
005795  1313  E003
005796                          *       WAIT
005797                          *       WAIT    ERROR, BVF              SHOULDN'T BRANCH
005798  1314  0101
005799                          *       GNB
005800                          *       DEC
005801  1315  0205
005802                          *       BZF     BVT                     BRANCH IF MORE CCB'S
005805  1316  F2F3
005806                          *       WAIT    WAIT                    FOR NEXT START IO
005807                          *  CCB LIST RESET HAS NOW RESET VALID BITS
005808                          *       LD      =4                      COUNTER
005809  1317  0190
005812                          *       LOC     BVF
005813          0213            BVF EQU X'0213'
005814                          *       DATA    X'F7'                   BZF
005817  1318  04F7
005818                          *       DATA    X'03'                   +3
```

```
005821                          *       WAIT
005822  1319  0301              *       WAIT    ERROR, BVF              DIDN'T BRANCH
005823                          *       DATA    X'E7'                   BVI
005824
005827  131A  01E7              *       DATA    X'3'                    +3
005828                          *       DATA    X'E0'                   B
005831
005834  131B  03E0              *       DATA    X'3'                    +3
005835                          *       WAIT
005838
005839  131C  0301              *       WAIT    ERROR, BVI              SHOULDN'T BRANCH
005840                          *       GNB
005841
005842  131D  0102              *       DEC
005843                          *       BZF     BVF                     MORE TO GO
005844
005845  131E  05F2              *       WAIT    DONE
005848
005849  131F  F301              *       WAIT
005850
005851        1320              ECP     EQU     $
005852                          *
005853                          *       DEFINED CONSTANTS
005854                          *
005855                          *
005856                          *
005857                          *
005858  1320  A001              CRC16   DC      Z'A001'
005859  1321  8408              CCITT   DC      Z'8408'
005860  1322  0000              LRC8    RESV    1,0
005861  1323  F010              CRC12   DC      Z'F010'
005862  1324  FFFF              ALLONE  DC      Z'FFFF'                 MINUS ONE OR ALL BITS SET CONSTANT
005863  1325  0010              C16     DC      X'10'                   DECIMAL 16 CONSTANT
005864  1326  0000              FRST    DC      0                       FRST TIME THRU FLAG
005865  1327  0000              LOOP    DC      0                       LOOP COUNTER
005866  1328  0000              PASS    DC      0                       PASS COUNT
005867  1329  FFFF              ATLT    RESV    16,-1                   ACTIVE LINE TABLE
005868  1339  0000              IMASK   RESV    1,0                     MASK OF ACTIVE LINES
005869  133A  5555              DFLT    DC      =X'5555'                DEFAULT VALUE
005870  133B  805C              CRLF    DC      Z'805C'
005871  133C  2020              SPACE   DC      Z'2020'
005872  133D  0000              MLCADD  DC      0                       MLCP ADDRESS
005873  133E  0000              FW-REV  DC      0                       MLCP FIRMWARE REV.
005874                          *
005875                          *
005876                          *
005877                          *               STORAGE CONTANTS
005878                          *
005879                          *
005880                          *
005881                          *
005882  133F  1341              ADDS    DC      <DUMMY                  ADDSRESS STORAGE
005883  1340  0400              BUFRNG  DC      X'400'                  LENGTH OF SEND, RECEIVE BUFFER
005884  1341  0000              DUMMY   RESV    1,0
005885  1342  0000              COUNT   RESV    1,0                     COUNTER
005886  1343  0000              CH-CT   DC      0                       CHANNEL COUNT
005887  1344  26C0              ENDSAV  DC      <LS1
005888  1345  0000              TIMEIN  DC      0                       INITIAL VALUE FOR PASS TIME
005889  1346  0000              FLAG    DC      0                       FLAG
005890  1347  0000              TPR     RESV    $AF,0                   TEMPORARY REGISTER
005891  1348  0000              TEMP    RESV    $AF,0
005892  1349  0000              TEMP1   RESV    $AF,0
005893  134A  0000              TEMPST  RESV    1,0                     STATUS STORED HERE
005894  134B  0000              SCFLG   RESV    1,0                     SCAN FLAG
005895  134C  0000              CCBFLG  RESV    1,0                     SET CCB AFTER RUPT FLAG
005896  134D  0000              QFLG    RESV    1,0
005897  134E  0000              RFLG    RESV    1,0
005898  134F  0000              PRTFLG  RESV    1,0                     FLAG TO PRINT TEST LABLE
005899  1350  0000              CRCAC   RESV    1,0                     CRC ACCUMULATOR
005900  1351  0000              IRG     RESV    1,0                     RANGE INPUT VALUE
005901  1352  0000              RANGE   RESV    1,0                     RANGE VALUE
005902  1353  0000              RANGE1  RESV    1,0                     SECONDARY RANGE
005903  1354  0000              RANGEW  RESV    1,0                     RANGE IN WORDS
005904  1355  0000              RAMAD   RESV    1,0                     RAM ADDRESS IN MLCC
005905  1356  0000              RAMAD2  RESV    1,0                     SECONDARY RAM ADDRESS
005906  1357  0000              IMGA    RESV    $AF,0                   DUMMY CCB ADDRESS
005907  1358  0000              TSA2    DC      0                       0 TO END TRAP SAVE AREA
005908  1359  13B4              SBCK    DC      <RMINST
005909  135A  0000              SBCCB   DC      0                       GETS SBCK WITH BYTES REVERSED
005910  135B  0008              SBCCP   DC      Z'0008'
005911  135C  00AA                      DC      Z'00AA'
005912  135D  5555                      DC      Z'5555'
005913  135E  0000              MASK    RESV    1,0                     MASK WORD
005914  135F  0000              MASK1   RESV    1,0                     SECONDARY MASK WORD
005915  1360  0000              ERAR    RESV    4,0                     ERROR ARRAY
005916  1364  0000              SAV1    RESV    9+7*$AF,0               SUB-PROGRAM REGS. SAVE AREA
005917  1374  0000              SAV2    RESV    9+7*$AF,0
005918  1384  0000              SAV3    RESV    9+7*$AF,0
005919  1394  0000              SAV4    RESV    9+5*$AF,0
005920  13A2  0000              SAV6    RESV    2*$AF,0
005921  13A4  0000              SAV5    RESV    9+7*$AF,0
005922  13B4  0000              RMINST  RESV    1,0                     RAM INSTRUCTION TO TEST
005923  13B5  0000              RTCHN   DC      0                       INTERRUPT RETURN CHANNEL
005924  13B6  0000              SEC     DC      0                       SECONDS STORED HERE
005925  13B7  0000              TTOT    DC      0                       TICK TOTAL
005926  13B8  0000              INTM    DC      0                       TEMPORARY TIME VALUE
005927  13B9  002D              ERF     DC      45                      ERROR FLAG 0 = SHORT PRINTOUT
005928  13BA  0000              STAT    DC      0                       LAST STATUS INPUT STORED HERE
005929  13BB  003C              HRTZ    DC      =60                     PWR FREQ
005930                          *
005931                          *       I/O CONTROL WORDS
005932                          *
005933  13BC  0009              CONT1   DC      Z'0009'                 IOLD FUNCTION CODE
005934  13BD  000C              CONT2   DC      Z'000C'                 INPUT RANGE FUNCTION CODE
005935  13BE  000F              CONT3   DC      Z'000F'                 OUTPUT CCB CONTROL FUNCTION CODE
005936  13BF  001A              CONT4   DC      X'1A'                   INPUT NEXT STATUS FUNCTION CODE
005937  13C0  000B              CONT5   DC      Z'000B'                 OUTPUT BYTE INTO LCT FUNCTION CODE
005938  13C1  0018              CONT6   DC      Z'0018'                 INPUT STATUS FUNCTION CODE
005939  13C2  0005              CONT7   DC      X'5'                    OUTPUT CHANNEL CONTROL FUNTION CODE
005940  13C3  0001              CONT9   DC      Z'0001'                 OUTPUT MLCC CONTROL FUNTION CODE
005941  13C4  001E              CONT10  DC      X'1E'                   INPUT LCT STATUS
005942  13C5  0003              CONT11  DC      X'3'                    OUTPUT INTERRUPT CONTROL FUNCTION CODE
005943  13C6  001C              CONT12  DC      X'1C'                   INPUT DATA SET STATUS FC
```

```
005944                                  *
005945                                  *
005946                                  *
005947                                  *
005948                                  *
005949                                  *CONSOLE MESSAGES
005950                                  *
005951                                  *
005952  13C7 4348 414E 4E45             MESG1   TEXT    'CHANNEL $'
        13CA 4C20 2400
005953  13CC 204C 494E 4520             IDMSG   TEXT    ' LINE $'
        13CF 2400
005954  13D0 2049 4420 3D20             EQ      TEXT    ' ID =  $'
        13D3 2024
005955  13D4 4D42 2020 2024             ERMG    TEXT    'MB   $'
005956  13D7 4D4C 4343 204E             ERMG1   TEXT    'MLCC NOT ON THIS CHANNEL$'
        13DA 4F54 204F 4E20
             5448 4953 2043
             4841 4E4E 454C
             2400
005957                                  *
005958                                  * ROUTINE TO PRINT TEST LABLE OR BREAK
005959                                  *
005960  13E4 9F00 13D5                  PLB     STR     $R1,<ERMG+1
005961                                          CALL    ZV$BRK
        13E6 FBF0 0001
        13E8 D380 0000          X
005962  13EA 8980 0000          X               CMZ     <ZV$BKF
005963  13EC 0980 01C9                          BNE     <ROTBG                  IF SET GO TO NEXT
005964                                  *
005965  13EE 8980 134F                          CMZ     <PRTFLG
005966  13F0 0982                               BNE     >PLB1
005967  13F1 8384                               JMP     $B4
005968                                  *
005969  13F2 9F00 13FE                  PLB1    STR     $R1,<PLB3
005970                                          CALL    ZV$IC,PLB2              PRINT LABLE
        13F4 FBC0 0003
        13F6 D380 0000          X
        13F8 0F80
        13F9 13FB
005971  13FA 8384                               JMP     $B4
005972                                  *
005973  13FB 5445 5354 2020             PLB2    TEXT    'TEST  '
005974  13FE 2020 2424                  PLB3    TEXT    '  $$'
005975       1400                       CNTU    EQU     $
005976  2400                            LAF99   ORG     ZERO+X'2400'
005977       2400                       SDB     EQU     $                       SEND BLOCK AREA
005978  2800                            LAF98   ORG     ZERO+X'2800'
005979       2800                       RTB     EQU     $                       RETURN BLOCK AREA
005980  2600                                    ORG     ZERO+X'2600'
005981                                  * INTERRUPT SAVE AREA FOLLOWS. THIS AREA WILL BE OVER WRITTEN BY
005982                                  * DATA DURING DATA TRANSFERS BUT THE FOLLOWING SAVE AREAS ARE
005983                                  * RE-CREATED BEFORE TESTING INTERRUPTS
005984                                  *
005985                                  *
005986                                  * INTERRUPT SAVE AREAS - 16 FOR MLCC
005987                                  *
005988                                  * CHANNEL 1
005989  2600 0000                       IS1     RESV    $AF,0                   TRAP SAVE POINTER
005990  2601 0000                               RESV    1,0                     CHAN,LEVEL
005991  2602 7884                               RESV    1,X'7884'               SAVE R1,R2,R3,R4,I,B5
005992  2603 0000                               RESV    1,0                     RFU
005993  2604 0F91                               DC      <IHD1
005994  2605 FFFF                               DC      -1                      MAKE SURE PRIVLEGE BIT IS SI
005995  2606 0000                               RESV    5+$AF,0
005996                                  * CHANNEL 2
005997  2607 0000                               RESV    $AF,0                   TRAP SAVE POINTER
005998  2608 0000                               RESV    1,0                     CHAN,LEVEL
005999  2609 7884                               RESV    1,X'7884'               SAVE R1,R2,R3,R4,I,B5
006000  260A 0000                               RESV    1,0                     RFU
006001  260B 0F91                               DC      <IHD1
006002  260C FFFF                               DC      -1                      SET PRIVILEGE BIT
006003  260D 0000                               RESV    5+$AF,0
006004                                  * CHANNEL 3
006005  260E 0000                               RESV    $AF,0                   TRAP SAVE POINTER
006006  260F 0000                               RESV    1,0                     CHAN,LEVEL
006007  2610 7884                               RESV    1,X'7884'               SAVE R1,R2,R3,R4,I,B5
006008  2611 0000                               RESV    1,0                     RFU
006009  2612 0F91                               DC      <IHD1
006010  2613 FFFF                               DC      -1                      SET PRIVILEGE
006011  2614 0000                               RESV    5+$AF,0
006012                                  * CHANNEL 4
006013  2615 0000                               RESV    $AF,0                   TRAP SAVE POINTER
006014  2616 0000                               RESV    1,0                     CHAN,LEVEL
006015  2617 7884                               RESV    1,X'7884'               SAVE R1,R2,R3,R4,I,B5
006016  2618 0000                               RESV    1,0                     RFU
006017  2619 0F91                               DC      <IHD1
006018  261A FFFF                               DC      -1                      SET PRIVILEGE BIT
006019  261B 0000                               RESV    5+$AF,0
006020                                  * CHANNEL 5
006021  261C 0000                               RESV    $AF,0                   TRAP SAVE POINTER
006022  261D 0000                               RESV    1,0                     CHAN,LEVEL
006023  261E 7884                               RESV    1,X'7884'               SAVE R1,R2,R3,R4,I,B5
006024  261F 0000                               RESV    1,0                     RFU
006025  2620 0F91                               DC      <IHD1
006026  2621 FFFF                               DC      -1                      SET PRIVILEGE BIT
006027  2622 0000                               RESV    5+$AF,0
006028                                  * CHANNEL 6
006029  2623 0000                               RESV    $AF,0                   TRAP SAVE POINTER
006030  2624 0000                               RESV    1,0                     CHAN,LEVEL
006031  2625 7884                               RESV    1,X'7884'               SAVE R1,R2,R3,R4,I,B5
006032  2626 0000                               RESV    1,0                     RFU
006033  2627 0F91                               DC      <IHD1
006034  2628 FFFF                               DC      -1                      SET PRIVILEGE BIT
006035  2629 0000                               RESV    5+$AF,0
006036                                  * CHANNEL 7
006037  262A 0000                               RESV    $AF,0                   TRAP SAVE POINTER
006038  262B 0000                               RESV    1,0                     CHAN,LEVEL
006039  262C 7884                               RESV    1,X'7884'               SAVE R1,R2,R3,R4,I,B5
006040  262D 0000                               RESV    1,0                     RFU
006041  262E 0F91                               DC      <IHD1
006042  262F FFFF                               DC      -1                      SET PRIVILEGE BIT
006043  2630 0000                               RESV    5+$AF,0
```

```
006044
006045  2654  0000                  * CHANNEL 8
006046  2655  0000                        RESV    $AF,0                   TRAP SAVE POINTER
006047  2656  7884                        RESV    1,0                     CHAN,LEVEL
006048  2657  0000                        RESV    1,X'7884'               SAVE R1,R2,R3,R4,1,B5
006049  2658  0F91                        RESV    1,0                     RFU
006050  2659  FFFF                        DC      <IHD1
006051  265A  0000                        DC      -1                      SET PRIVILEGE BIT
006052                                    RESV    5+$AF,0
006053  265B  0000                  * CHANNEL 9
006054  265C  0000                        RESV    $AF,0                   TRAP SAVE POINTER
006055  265D  7884                        RESV    1,0                     CHAN,LEVEL
006056  265E  0000                        RESV    1,X'7884'               SAVE R1,R2,R3,R4,1,B5
006057  265F  0F91                        RESV    1,0                     RFU
006058  2660  FFFF                        DC      <IHD1
006059  2661  0000                        DC      -1                      SET PRIVILEGE BIT
006060                                    RESV    5+$AF,0
006061  2662  0000                  * CHANNEL 10
006062  2663  0000                        RESV    $AF,0                   TRAP SAVE POINTER
006063  2664  7884                        RESV    1,0                     CHAN,LEVEL
006064  2665  0000                        RESV    1,X'7884'               SAVE R1,R2,R3,R4,1,B5
006065  2666  0F91                        RESV    1,0                     RFU
006066  2667  FFFF                        DC      <IHD1
006067  2668  0000                        DC      -1                      SET PRIVILEGE BIT
006068                                    RESV    5+$AF,0
006069  2669  0000                  * CHANNEL 11
006070  266A  0000                        RESV    $AF,0                   TRAP SAVE POINTER
006071  266B  7884                        RESV    1,0                     CHAN,LEVEL
006072  266C  0000                        RESV    1,X'7884'               SAVE R1,R2,R3,R4,1,B5
006073  266D  0F91                        RESV    1,0                     RFU
006074  266E  FFFF                        DC      <IHD1
006075  266F  0000                        DC      -1                      SRT PRIVILEGE BIT
006076                                    RESV    5+$AF,0
006077  2670  0000                  * CHANNEL 12
006078  2671  0000                        RESV    $AF,0                   TRAP SAVE POINTER
006079  2672  7884                        RESV    1,0                     CHAN,LEVEL
006080  2673  0000                        RESV    1,X'7884'               SAVE R1,R2,R3,R4,1,B5
006081  2674  0F91                        RESV    1,0                     RFU
006082  2675  FFFF                        DC      <IHD1
006083  2676  0000                        DC      -1                      SET PRIVILEGE BIT
006084                                    RESV    5+$AF,0
006085  2677  0000                  * CHANNEL 13
006086  2678  0000                        RESV    $AF,0                   TRAP SAVE POINTER
006087  2679  7884                        RESV    1,0                     CHAN,LEVEL
006088  267A  0000                        RESV    1,X'7884'               SAVE R1,R2,R3,R4,1,B5
006089  267B  0F91                        RESV    1,0                     RFU
006090  267C  FFFF                        DC      <IHD1
006091  267D  0000                        DC      -1                      SET PRIVILEGE BIT
006092                                    RESV    5+$AF,0
006093  267E  0000                  * CHANNEL 14
006094  267F  0000                        RESV    $AF,0                   TRAP SAVE POINTER
006095  2680  7884                        RESV    1,0                     CHAN,LEVEL
006096  2681  0000                        RESV    1,X'7884'               SAVE R1,R2,R3,R4,1,B5
006097  2682  0F91                        RESV    1,0                     RFU
006098  2683  FFFF                        DC      <IHD1
006099  2684  0000                        DC      -1                      SET PRIVILEGE BIT
006100                                    RESV    5+$AF,0
006101  2685  0000                  * CHANNEL 15
006102  2686  0000                        RESV    $AF,0                   TRAP SAVE POINTER
006103  2687  7884                        RESV    1,0                     CHAN,LEVEL
006104  2688  0000                        RESV    1,X'7884'               SAVE R1,R2,R3,R4,1,B5
006105  2689  0F91                        RESV    1,0                     RFU
006106  268A  FFFF                        DC      <IHD1
006107  268B  0000                        DC      -1                      SET PRIVILEGE BIT
006108                                    RESV    5+$AF,0
006109  268C  0000                  * CHANNEL 16
006110  268D  0000                        RESV    $AF,0                   TRAP SAVE POINTER
006111  268E  7884                        RESV    1,0                     CHAN,<LEVEL
006112  268F  0000                        RESV    1,X'7884'               SAVE R1,<R2,<R3,<R4,<1,B5
006113  2690  0F91                        RESV    1,0                     RFU
006114  2691  FFFF                        DC      <IHD1
006115  2692  0000                        DC      -1                      SET PRIVILEGE BIT
006116        2696                        RESV    5+$AF,0
006117                              LS1   EQU     $
006118                              *
006119                              * INTERRUPT SAVE AREA - LEV 0
006120                              *
006121  2696  0000                        RESV    $AF,0                   TRAP SAVE POINTER
006122  2697  0000                  TOPSAV RESV   1,0                     CHAN, LEVEL
006123  2698  FFFF                        DC      Z'FFFF'                 SAVE ALL REG
006124  2699  0000                        RESV    1,0                     RFU
006125  269A  1341                        DC      <DUMMY                  SUPPLIED AT RUPT TIME
006126  269B  FFFF                        DC      -1                      SET PRIVILEGE BIT
006127  269C  0000                        DC      0                       FILLED AT SAVE TIME
006128  269D  0000                        RESV    8+7*$AF,0               ROOM FOR SAVE
006129                              *
006130                              * COUNT TABLE FOR INTERRUPTS
006131                              *
006132  26C5  0000                  INMB  RESV    16,0
006133                              *
006134                              * STATUS TABLE FOR INTERRUPTS
006135                              *
006136  26D5  0000                  IST   RESV    16,0
006137                              *
006138                              * LEVEL TABLE FOR INTERRUPTS
006139                              *
006140  26E5  0000                  ILV   RESV    16,0
006141                              *
006142                              * PRIORITY TABLE FOR INTERRUPTS
006143  26F5  0000                  PRIST RESV    16,0
006144  2705  1400                        ORG     CNIU
006145                              *
006146                              * READ FIRMWARE REV NUMBER
006147  1400  8F00 2400              *
        1402  0008                  PREV  SAVE    <SDB,=Z'0008'           SAVE B4
006148  1403  C380 106D                    LNJ     $B4,<GEN11Z             GENERAL INITAILIZE
006149  1405  9380 1009                    LNJ     $B1,<RDATA              BLOCK READ
006150  1407  2800                        DC      <RTB                    TO HERE
006151  1408  0002                        DC      2                       RANGE
006152  1409  0001                        DC      1
006153  140A  0000                        DC      0                       EVEN BYTE START
006154                              *
006155                              *
```

```
006156                                          CALL    ZV$IC,FREV          PRINT FIRMWARE REV
        140B  FBC0 0003
        140D  D380 0000       X
        140F  0F80
        1410  1450
006157  1411  9280 2800                        LLH     $R1,<RTB            GET REV NUMBER
006158  1413  9F00 133E                        STR     $R1,<FW-REV
006159  1415  9570 00E0                        AND     $R1,=Z'00E0'
006160  1417  1045                            SOR     $R1,5               ALIGN OPTION TYPE
006161  1418  9F00 1348                        STR     $R1,<TEMP
006162                                          CALL    ZV$ID,TEMP          TYPE OPTION
        141A  FBC0 0003
        141C  D380 0000       X
        141E  0F80
        141F  1348
006163                                          CALL    ZV$I,DASH           TYPE DASH
        1420  FBC0 0003
        1422  D380 0000       X
        1424  0F80
        1425  146E
006164                                  *
006165  1426  9280 2800                SIT12   LLH     $R1,<RTB            GET REV NUMB
006166  1428  9570 001F                        AND     $R1,=Z'001F'        STRIP TO REV
006167  142A  9F00 1348                        STR     $R1,<TEMP           PUT BACK
006168                                          CALL    ZV$ID,TEMP          PRINT NUMBER
        142C  FBC0 0003
        142E  D380 0000       X
        1430  0F80
        1431  1348
006169  1432  6C08                            LDV     $R6,=8
006170  1433  D280 2800                        LLH     $R5,<RTB
006171  1435  D956                            CMR     $R5,=$R6
006172  1436  0289                            BGE     >BLAP
006173                                  WFW     CALL    ZV$IC,ERMG2         FW REV WRONG
        1437  FBC0 0003
        1439  D380 0000       X
        143B  0F80
        143C  1443
006174                                  *                                   REV F OF PROGRAM DOESN'T SUPPORT
006175  143D  0F80 0105                        B       <RESTRT
006176  143F  8F80 2400                BLAP    RSTR    <SDB,=Z'0008'       RESTORE B4
        1441  0008
006177  1442  8384                            JMP     $B4
006178  1443  4D4C 4350 2046          ERMG2   TEXT    'MLCP FW REV NOT SUPPORTED$'
        1446  5720 5245 5620
              4E4F 5420 5355
              5050 4F52 5445
              4424
006179  1450  4D4C 4350 204F          FREV    TEXT    'MLCP OPT/FIRMWARE REV$'
        1453  5054 2F46 4952
              4D57 4152 4520
              5245 5624
006180  145B  4D4C 4350 2054          MESG    TEXT    'MLCP TEST '
        145E  4553 5420
006181                                          IFZ     ($AF-Z),LAF8
006182  1460  4D4C 4353 3120                    TEXT    'MLCS1  REV F  '
        1463  2052 4556 2046
              2020
006185  1467  204A 554E 4520          DATE    TEXT    ' JUNE 16 1978$'
        146A  3136 2031 3937
              3824
006186  146E  202D 2024                DASH    TEXT    ' - $'
006187  1470  5057 5220 4652          MESG5   TEXT    'PWR FREQ (HZ) $'
        1473  4551 2028 687A
              2920 2400
006188  1478  434F 5059 5249                    TEXT    'COPYRIGHT 1978 BY HONEYWELL INFORMATION SYSTEMS INC.'
        147B  4748 5420 3139
              3738 2042 5920
              484F 4E45 5957
              454C 4C20 494E
              464F 524D 4154
              494F 4E20 5359
              5354 454D 5320
              494E 432E
006189                                  *
006190                                  *
006191                                  *
006192                                  *
006193  1492  0100                            END     MLCS1,STRT
```

0000 ERR COUNT

TITLE MLCS1, 'REV F' MLCP TEST (SAF)

```
      $AF      456B    593C    611B    662B    666B    672B    688B    693B    697B    703B
               707B    719B    737     780B    790B   1021B   1026B   1181B   1194B   1395B
              1407    1432    1461B   1470B   1620B   1640B   1664B   1668B   1672B   1708B
              1740B   1756B   1812B   1848B   1873B   1878B   1882B   1932B   1940B   2206B
              2258B   2269B   2343B   2376B   2389B   2524B   2533B   2549B   2700C   2701C
              2771    3063B   3195B   3221B   3261B   3292B   3359B   3363B   3378B   3406B
              3421B   3431B   3499C   3602B   3613    3622    3631    3640    3646    3690
              3710    3730    3731C   3732    3733C   3734    3735C   3795C   3799    3946
              3962B   3965B   4035B   4046    4048    4092B   4158C   4181    4225B   4288B
              4293B   4305B   4399C   4458    4459C   4463    4464    4468    4474    4475
              4479    4481    4486C   4489    4495    4504    4505    5890    5891    5892
              5906    5916    5917    5918    5919    5920    5921    5989    5995    5997
              6003    6005    6011    6013    6019    6021    6027    6029    6035    6037
              6043    6045    6051    6053    6059    6061    6067    6069    6075    6077
              6083    6085    6091    6093    6099    6101    6107    6109    6115    6120
              6127    6181    6183
      $B1      569B    852B    866B    874B    881B    886B    893B    896B    903B    929B
               937B    940B    947B    960B    967B    970B    977B    984B    998B   1160B
              1173    1176    1228B   1236B   1252    1254    1275B   1284B   1369B   1375B
              1507B   1514B   1521B   1527B   1533B   1539B   1548B   1569B   1631B   1701B
              1703B   1719B   1745B   1746B   1747B   1760B   1761B   1774B   1781B   1793B
              1834B   1866B   1885B   1896B   2182    2183C   2233B   2325    2326C   2400B
              2464B   2580    2587C   2590C   2592    2594C   2633B   2648B   2654B   2662B
              2669B   2784B   2807B   2882B   2995B   3011B   3040    3041C   3071B   3096
              3097C   3109B   3155B   3253B   3259    3260C   3317B   3333    3334C   3385B
              3447B   3483B   3512B   3542B   3547B   3588B   3775    3780    3781C   3785
              3797B   3818    3819    3821    3828B   3843B   3851    3854    3937    3938
              3940    3942    3968B   3983    3985    3986    3988    4326    4332C   4474
              4479    4480C   4509C   6149B
      $B2      541B    542B    543B    544B    545B    546B    547B    548B    549B    550B
               551B    552B    553B    554B    555B    556B    557B    558B    683B    748B
               829B   1013B   1040B   1056B   1202B   1217B   1298B   1361B   1463B   1492B
```

```
                1612B   1655B   1908B   2057B   2147B   2444    2447C   2451C   2455C   2503
                2626B   2799B   2899B   2994B   3149B   3594B   3703    3707C   3711    3713C
                3721    3722C   3724    3725C   3726    3728C   3731C   3733C   3735C   3821
                3822    4073    4076    4281    4287    4463    4467C   4468    4470C   4494C
                4498C   4505    4508C
$B3              854     857C   1122B   1127    1130    1144    1145    1147C   1244B   1250B
                1382B   1386B   1400B   1405B   1428B   1430B   1446B   1472B   1556B   1558B
                1697B   1751B   1789B   1795B   1803B   1911B   2453    2454C   2458    2459C
                2471B   2483B   2518    2522    2642    2692    2694    2696    2697C   2705B
                2735    2736C   2742B   2783    2817    2824B   2848B   2875    2999B   3049B
                3118B   3168B   3232B   3271B   3302B   3309B   3472    3473C   3558B   3566B
                3570B   3706    3707C   3772    3775    3776    3778C   3779    3781C   3783C
                3785    3786    3786    3787    3822    3824C   3851    3858B   4007    4008
                4012B   4041B   4048    4054C   4160    4180C   4187    4204C   4325    4327
                4329    4473    4480C   4485    4486C   4509C
$B4              411     412C    413     414C    454B    475B    515B    588B    590B    596B
                 604B    615B    623B    652B    656B    660B    682B    686B    691B    701B
                 717B    733B    735B    747B    765B    772B    778B    785B    787B    805B
                 821B    823B    828B   1008B   1010B   1012B   1019B   1024B   1029B   1033B
                1035B   1051B   1052B   1055B   1063B   1093B   1135B   1136B   1151B   1189B
                1211B   1213B   1216B   1251B   1255B   1256B   1265B   1285B   1290B   1297B
                1350B   1352B   1360B   1385B   1388B   1390B   1409B   1413B   1414B   1438B
                1452B   1453B   1455B   1468B   1488B   1491B   1495B   1561B   1609B   1611B
                1616B   1618B   1652B   1654B   1657B   1659B   1662B   1675B   1681B   1689B
                1706B   1710B   1738B   1754B   1765B   1775B   1779B   1796B   1797B   1804B
                1806B   1809B   1858B   1864B   1865B   1871B   1890B   1904B   1905B   1907B
                1910B   1914B   1920B   1924B   1928B   1937B   1945B   1948B   1966B   1967B
                1970B   1971B   1975B   1987B   1988B   2000B   2001B   2015B   2027B   2028B
                2042B   2043B   2055B   2056B   2076B   2077B   2090B   2091B   2104B   2105B
                2116B   2119B   2132B   2133B   2145B   2146B   2169    2170    2176    2179
                2182    2185    2204B   2220B   2222B   2243B   2254B   2266B   2281B   2282B
                2293B   2308    2309    2314    2318    2325    2328    2341B   2349B   2351B
                2356B   2368B   2386B   2417B   2418B   2426B   2444    2445    2453    2456
                2458    2460C   2477B   2485B   2487    2489B   2503    2504    2538B   2583
                2584    2586    2597B   2605B   2606B   2616B   2686B   2729B   2745B   2748B
                2763B   2778B   2793B   2796B   2798B   2822B   2828B   2833B   2838B   2843B
                2852B   2861B   2865B   2879B   2898B   2989B   2992B   2993B   2996B   2997B
                3022B   3028B   3032B   3035B   3042B   3061B   3086B   3093B   3098B   3114B
                3146B   3148B   3150B   3151B   3191B   3192B   3193B   3204B   3214B   3215B
                3219B   3241B   3246B   3250B   3252B   3261B   3279B   3287B   3289B   3299B
                3300B   3316B   3320B   3329B   3335B   3347B   3356B   3357B   3371B   3375B
                3398C   3404B   3408C   3409B   3417C   3419B   3423C   3424B   3427C   3429B
                3433C   3434B   3457B   3467B   3482B   3487B   3493B   3554B   3560B   3574B
                3579B   3600B   3605B   3861B   3945B   3948B   3955B   3959B   3960B   3991B
                3994B   4022B   4033B   4057B   4079    4081B   4087B   4090B   4094B   4102B
                4185B   4223B   4229B   4246B   4255    4256    4258B   4264B   4281    4282
                4283C   4286B   4291B   4295C   4297B   4301C   4303B   4308C   4309B   4325
                4326    4335B   4349    4350    4354B   4361B   4369B   4372B   4382B   4401B
                4407B   4433B   4440    4443C   4444B   4445    4449B   4462    4469    4494C
                4507    4508C   5967B   5971B   6148B   6177B
$B5              482     486     488C   2274B   2410B   2460C   2474    2475    2487    2515
                2516    2551B   2610    2611C   2689    2690C   2694    2699C   2700C   2701C
                2703    2704C   2771    3398C   3408C   3417C   3423C   3427C   3433C   3702
                3703    3704    3706    3710    3713C   3719    3722C   3727    3728C   3730
                3731C   3732    3733C   3734    3735C   3746    3747C   3911    3912    3915C
                3916C   3937    3939C   3983    3984C   4046    4048    4052C   4054C   4283C
                4295C   4301C   4308C   4344    4398    4399C   4466    4467C
$B6              836B    838B    840B    842B    980B   2519    2521    2530    2641    2818
                2876    4151    4458    4459C   4461    4470C   4497    4498C
$B7              457B    607B    612B    616B    626B    647B    663B    667B    673B    689B
                 694B    698B    704B    706B    721B    781B    791B    801B    996B   1020
                1022B   1027B   1039B   1140B   1162B   1196B   1258B   1269B   1396B   1418B
                1423B   1462B   1471B   1581B   1621B   1641B   1665B   1669B   1673B   1709B
                1730B   1741B   1757B   1813B   1849B   1874B   1879B   1883B   1933B   1941B
                2259B   2270B   2287B   2377B   2390B   2423B   2525B   2534B   2550B   2758B
                2775B   2858B   2895B   3064B   3196B   3222B   3282B   3293B   3360B   3364B
                3379B   3407B   3422B   3432B   3603B   3662B   3669B   3839B   3878B   3886B
                3890B   3894B   3903B   3963B   3966B   4036B   4093B   4226B   4261B   4289B
                4294B   4306B   4364B   4458    4503B
$IV             3873    3879
$R1              391     392C    395     396C    397     398     405     406     410     415
                 418     419     441     444     452     474     484     485C    487     488C
                 497     498     499     501     506     508     512     518     560     561
                 587     592     593C    594C    658     670B    684     712B    732     737
                 738     739     740C    741C    820     834C    926     927B   1007    1014C
                1050    1069C   1072    1073C   1074C   1079C   1080    1081C   1088    1089C
                1090    1091C   1130    1131C   1132    1133B   1134    1170C   1175    1176
                1179    1210    1254    1282    1283C   1349    1356    1357    1364C   1406
                1408B   1431    1433B   1487    1498C   1608    1651    1677    1678C   1726
                1732B   1903    1965    1969    1986    1999    2013    2014C   2026    2041
                2054    2075    2089    2103    2117    2131    2144    2167    2168C   2169
                2170    2171C   2172C   2173C   2174C   2176    2177C   2179    2180C   2185
                2186C   2187C   2188C   2190    2199    2200    2201C   2280    2308    2309
                2310C   2311C   2312C   2313C   2314    2315C   2318    2319    2322C   2328
                2329C   2330    2332    2333C   2334C   2335    2336    2337    2338C   2416
                2442    2443C   2456    2457C   2474    2505    2506C   2515    2521    2522
                2530    2586    2587C   2590C   2591    2595B   2604    2621    2622C   2623C
                2624    2628C   2691    2692    2728C   2734    2735    2762C   2792    2795C
                2988    3003    3004C   3006    3007C   3020C   3072    3076    3078    3079C
                3081    3082C   3084    3085C   3092C   3110    3111B   3145    3188    3190
                3200    3202    3203C   3227    3228C   3234    3235    3237C   3238    3240
                3247    3249    3254    3255C   3286    3318    3319C   3321    3322C   3323
                3324C   3328    3344    3346    3368    3370    3399    3400    3401C   3402
                3405    3420    3438    3442    3445    3446C   3466    3490C   3507    3508
                3509B   3511C   3525    3526    3529C   3533B   3565    3575B   3712    3714B
                3718    3723B   3736C   3737C   3738C   3739C   3740C   3741C   3742C   3743
                3744C   3773C   3776    3793    3794C   3795C   3820    3824C   3826B   3852C
                3854    3855C   3871    3872    3891    3942    3943C   3944    3986    3987C
                3988    3989C   3990    4030    4031B   4034    4049    4050    4052C   4152C
                4168C   4170    4172B   4191C   4193    4196    4287    4329    4330    4331
                4345C   4346C   4347    4348C   4349    4350    4351C   4352C   4367    4368
                4375    4376C   4377    4378    4389    4390    4396    4397    4400C   4402C
                4403C   4404    4405C   4409    4411B   4412    4422C   4440    4441C   4442C
                4446    4447C   4456    4499B   5960C   5969C   6157    6158C   6159    6160
                6161C   6165    6166    6167C
$R2              481     486     488C   489B    653     661     668C    671     679     695
                 710C    847    1070    1071    1075    1080    1082B   1083C   1128    1130
                1141B   1143    1145    1147C   1148B   1253C   1254    1259    1660C   1667
                1727    1728    1731    1845    1846    1850C   1851C   1854    1856    1961
                1963    1965    2317C   2321    2323C   2448C   2449C   2450    2451C   2475
                2476C   2481    2516    2517C   2526B   2583    2587C   2588C   2593    2594C
                2612    2613C   2688    2689    2702    2703    3477    3502C   3503B   3506
                3507    3520    3521B   3526C   3530    3532    3535C   3536C   3537C   3818
```

```
                3824C   3834    3841B   3854    3856B   3862    3898    3900    3940    3941C
                4045C   4049    4053C   4055    4072    4076    4079    4156C   4170    4171
                4173    4177    4180C   4183C   4193    4194    4195    4198    4202    4204C
                4205    4208    4214    4282    4292
        $R3      441     442     443     445C    446C    458C    459B    461     462C    464
                 470     471C    472     514C    595C    638C    655C    674C    681C    723C
                 734C    749C    750C    751C    752C    754     757C    758B    759C    761C
                 770     776     792     807C    827C    831B    832C    843C    861C    862C
                 863B    865C    873C    885C    895C    911     925C    939C    959C    969C
                 979     1009C   1054C   1057C   1058B   1059C   1188B   1198    1199C   1200
                1215C   1224C   1225B   1226C   1249    1271    1272B   1273C   1289C   1295
                1296C   1351C   1362B   1363C   1365C   1437C   1445C   1489C   1493C   1494C
                1496B   1497C   1563    1567C   1582    1583C   1610C   1623    1627C   1628B
                1629C   1637    1642C   1653C   1682    1712C   1713    1714    1715C   1716
                1717    1733    1814C   1827    1830C   1831B   1832C   1840    1853B   1906C
                1950C   1962B   2203C   2208B   2209C   2211    2215C   2231C   2239C   2288C
                2340C   2345B   2346C   2348C   2394    2399C   2406C   2424C   2461C   2462B
                2463C   2470    2540    2541B   2584    2585C   2589B   2614C   2618B   2619C
                2629C   2640C   2671    2794C   2800B   2801C   2867    2881C   2897    2900B
                2901C   2990C   3004C   3007C   3009C   3010    3016    3077B   3079C   3082C
                3085C   3125C   3147C   3153C   3154    3161    3189B   3201B   3239B   3255C
                3319C   3322C   3345B   3369B   3389C   3439C   3440B   3441C   3443    3456
                3476C   3498C   3528    3531B   3559C   3819    3823B   3825    3833C   3835
                3837    3840C   3873    3874    3875    3879    3880    3881C   3882    3883
                3891    3900    3915C   3918C   4009    4010B   4015    4019    4029C   4030
                4038C   4039    4074    4076    4078C   4086C   4155C   4160    4167    4200
                4201    4203    4206    4207    4209    4243    4244    4245    4457C   4473
                4476    4484C   4485    4493C   4495
        $R4      420     440     452     453     455     474     476     483     486     487
                 600     601     602C    603     605     609     610     614     616     622
                 624     659     661     664     665     671     685     687     690     692
                 700     702     716     718     743     744C    763     771     777     779
                 797     825     826C    835C    837     839     841     857C    858     882
                 883C    911     912     913C    914C    915     916C    917C    918     919C
                 920C    921C    922     923C    955     956     957C    992     1018    1020
                1023    1025    1071    1072    1073C   1078    1083C   1118    1119C   1120
                1121C   1145    1146    1147C   1155    1156B   1157    1158C   1171    1183B
                1393C   1459C   1467    1469    1499    1500C   1563    1564    1565    1566C
                1617    1619    1623    1624    1625    1626C   1658    1660C   1661    1663
                1671    1690    1691C   1692C   1693C   1694C   1695C   1696C   1705    1707
                1737    1739    1753    1755    1772    1773C   1827    1828    1829C   1870
                1872    1876    1877    1889    1893C   1894    1913    2197    2198C   2211
                2212    2213    2214C   2394    2395C   2396    2397    2398C   2445    2448C
                2504    2581    2582C   2590C   2727    2733    2734    2747    2750    2761
                2827    2832    2837    2842    2860    3019    3034    3036C   3091    3113
                3403    3405    3418    3420    3428    3430    3443    3444    3445    3474
                3475C   3478    3599    3601    3777    3778C   3779    3782    3783C   3860
                3862    3881C   3897    3898    3938    3985    4014    4023    4032    4034
                4089    4091    4151    4164    4166    4173    4198    4222    4224    4244
                4257    4259    4284    4285    4287    4290    4292    4302    4304    4327
                4328    4331    4332C   4360    4362    4406    4413    4416    4417    4427
        $R5      419     420     421     422     428C    440     445C    447     455     458C
                 466     467     468C    476     624     627     692     696     702     706
                 718     745     746C    754     755     756     760     762C    789     798
                 949     950     993     1015    1016C   1020    1037    1068    1076B   1138
                1150    1175    1180    1193    1257B   1267    1394    1416    1421    1460
                1576    1577    1579    1638    1639    1728    1729B   1739    1811    1846
                1847    1877    1881    1931    1939    1968C   1972    1973    2193    2194C
                2195C   2257    2267    2268    2283    2285    2373    2375    2387    2388
                2419    2421    2446    2447C   2451C   2521    2523    2530    2532    2548
                2643    2650    2656    2664    2749C   2750    2753    2772    2819    2856
                2874    2889    2890    2891    3053    3062    3194    3205    3206    3207C
                3208C   3209    3210    3213C   3220    3280    3291    3358    3362    3377
                3453    3454    3455C   3469    3470C   3491    3497B   3505    3516C   3517
                3519C   3592    3601    3961    3964    4007    4023    4159B   4160    4168C
                4169    4177    4178    4179C   4187    4190    4191C   4192    4205    4210
                4211C   4214    4215C   4224    4227C   4255    4256    4259    4284    4287
                4304    4307    4324C   4333B   4355    4362    4416    4427    4429C   6170
                6171
        $R6      599     605     627     633     636C    695     696     705     706     788
                 789     799     848     849C    855     856C    859B    994     1131C   1137
                1138    1176    1177    1180    1192C   1193    1266    1267    1392    1394
                1416    1420    1421    1458    1460    1578    1579    1613    1614    1615
                1619    1639    1810    1811    1844C   1847    1880    1881    1930    1938C
                1939    1960    1964    1969    2255    2256    2257    2260C   2262    2268
                2284C   2285    2367C   2370    2374    2375    2379C   2380    2388    2420C
                2421    2522    2523    2531    2532    2542    2545    2547    2548    2608
                2609C   2644    2645C   2646    2651    2652C   2653    2657    2658C   2659
                2660C   2661    2666    2667C   2668    2740C   2756B   2773    2820    2867
                2868    2869    2870C   2877    2893    3290    3291    3376    3377    4008
                4017B   4018    4157C   4158C   4174B   4181    4199B   4418    4419    6169
                6171
        $R7      654     665     669     680     705     711     800     995     1179    1851C
                2647    2726    2737B   2739    2759    2766B   2774    2782    2821    2878
                3203C   3206    3210    3943C   3954B   3989C   4018    4019    4020B   4162
                4163C   4182B
3494    ADD     3473C   3499C   3502C
5882    ADDS    2611C   2689    2690C   2697C   2703
1664    ADV2    1703B   1747B
 483    ALL      489B
5862    ALLONL   436    3765
5867    ATLT     436     445C    458C    4073
3057    BKHK    3046B
 541    BKK1     535B
 558    BKK10
 542    BKK3     540     567B
 543    BKK4
 544    BKK4A
 545    BKK4B
 547    BKK5
 548    BKK6
 549    BKK7
 550    BKK7A
 551    BKK7B
 552    BKK7C
 553    BKK7D
 554    BKK8
 555    BKK9
 556    BKK9A    521B    566B    573B
 557    BKK9B
6176    BLAP    6172B
 606    BLOC     604B
1827    BRLCLR  1681B   1775B
```

```
4542  BT1     4537   4538   4539
4591  BT11    4612   4613   4615
4594  BT12    4587   4588   4589
4606  BT17    4597   4598   4600
4609  BT18    4602   4603   4604
4623  BT1D    4616   4617   4621
4787  BT4B    4760   4761   4763   4772   4773   4775   4801   4802   4803
4734  BT4C    4706   4707   4708   4718   4719   4720   4748   4749   4751
4680  BT4D    4653   4654   4656   4665   4666   4668   4694   4695   4696
4642  BT4F    4637   4638   4639
5026  BT4G    5021   5022   5026
5024  BT4H    5030   5031   5032
4805  BT4J
4556  BT5     4582   4583   4585
4619  BT5C    4625   4626   4627
4560  BT6     4553   4554   4558
4942  BTA6    4843   4844   4845
4641  BTA7    4834   4835   4839
4968  BTA9    4952   4953   4954   4982   4983   4985   4994   4995   4997
4684  BTAC    4677   4678   4682
4576  BTB     4562   4563   4564
4971  BTB5    4964   4965   4966
4837  BTB7    4851   4852   4853
4579  BTC     4572   4573   4574
4737  BTRC    4730   4731   4732
4791  BTXC    4784   4785   4789
4823  BTYD
5883  BUFRNG  1221   2219
5613  BVF     5846   5847   5849
5771  BVT     5803   5804   5805
2427  BWENB   2344B
2361  BWLCT   2326C
5863  C16      436   3752   3756   3761   3765
1722  CADD    1715C
 767  CBRT1    762C   797
 761  CBRT3    756B
5895  CCBFLG  3738C
3487  CCBLP   3503B
 732  CCBRT    542B
2363  CCBSAV  2322C  2369   2371   2378   2543
5859  CCITT   1221
4243  CGSCH    454B   475B   604B   615B   623B   660B   686B   691B   701B   717B
          778B  1019B  1024B  1468B  1618B  1659B  1662B  1706B  1738B  1754B
         1871B  2748B  3035B  3404B  3419B  3429B  3600B  3861B  4022B  4033B
         4090B  4223B  4258B  4286B  4291B  4303B  4361B  4407B
5866  CH-CT    437C
3514  CHAN    3511C
4254  CHCT    1029B  1285B  1290B  1409B  1436B  1765B  1797B  1806B  1920B  1945B
         2778B  3948B  3994B
3427  CHINTZ  2996B  3299B
2521  CHK1    2526B
2547  CHK2    2541B
2548  CHK3    2544B  2546B
2515  CHKBW   2274B  2410B
1870  CHKRNG  1746B  1761B
3832  CHLV    3109B  3385B  3588B
3835  CHLV1   3841B
3840  CHLV2   3836B  3838B
1349  CHPGT    547B
1364  CHPGT2  1362B
1360  CHPGT3  1358B
2353  CHRG1   2334C
2358  CHRG2   2315C  2330   2332
2402  CHRG3   2333C
2412  CHRG4   2329C  2335
2354  CHST1   2310C
2362  CHST2   2311C
2404  CHST3   2312C
2411  CHST4   2313C
4263  CHZ     4260B
3799  CNFILL  3771
5975  CNTD    6143
2768  COMP    2700C  2771
 418  CON1     416B
 436  CON2     423B
 445  CON2A    477B
 471  CON2B    478B
 470  CON2C    448B   459B
 481  CON3     473B
5933  CONT1    482    603    659    1018   1658   4285
5941  CONT10  4302
5942  CONT11  1617   3403   4032
5943  CONT12  3599
5934  CONT2    622    690   3034
5935  CONT3   1023   4290
5936  CONT4    614    685    716   1737   1870   3897   4222
5937  CONT5   3418   3860
5938  CONT6    700   2747   4360   4406
5939  CONT7    777   1467   1705   1753   3428   4014   4257
5940  CONT9   1661   4089
5885  COUNT   2609C  2621   2623C
2730  CP1     2699C  2735   2736C
2706  CP2     2701C
2764  CP3     2704C
1474  CPT2    1401   1447
2676  CPT3    2706   2825
2679  CPT3A   2645C  2652C  2660C  2667C
1479  CPT4    1383
1478  CPT5    1429
4150  CRC     2686B  2822B  2879B
5861  CRC12
5858  CRC16   2641   2818   2876
4168  CRCA    4174B
4164  CRCAA   4182B
4163  CRCAA1  4159B
5899  CRCAC   2891   2893   4183C
4174  CRCB    4172B
4180  CRCC    4216B
4181  CRCC2   4188B  4212B
4190  CRCD    4167B  4199B
4187  CRCDD   4176B
4199  CRCE    4197B
4214  CRCF    4165B
```

```
2604  CRCT      554B
5870  CRLF     4461   4497
 989  CRWB     881B    903B    947B    977B   1284B
6186  DASH     6163
6185  DATE     6183
5869  DFLT      983   1679   2219   2708
4388  DLAY      785B  1033B  1251B  1385B  1388B  1413B  1453B  1561B  1710B  1796B
               1804B  1924B  1948B  2485B  2865B  2997B  3151B  3215B  3246B  3300B
               3356B  3574B  3959B
4411  DLAY1    4430B
4398  DLAY2    4391B
4426  DLAY3    4420B
4418  DLAY4    4411B
4430  DLAY5    4424B
4395  DLAYLG   1675B  3192B  4094B
3278  DOL2     3265B  3273
3355  DOL3     3339B
3443  DOL4     3440B
2485  DOLA     2482B
 390  DRSR      387B
 387  DRSR1     392C
3232  DSLOOP   3295B
3145  DSS1      557B
3598  DSSTA    1452B  3204B
3148  DSSTLP   3390B
1817  DTP1     1752
5884  DUMMY     404    405    616    687   2248   2361   2465   2472   2478   2730
               2764   2883   2889   3494   3652   3992   5882   6124
1745  ECO3
1751  ECO2     1742
5851  ECP      1783
5887  ENDSAV   3787
2403  EOADD    2398C
2352  EODATA   2348C
5954  EQ        465
4470  ER11     4464
5915  ERAR      794    795    798    799    800    989    990    993    994    995
               2768   2769   2772   2773   2774
5927  ERF      4471
5955  ERMG    2014C  4418   4460   5960C
5956  ERMG1     425
6178  ERMG2    6173
4460  ERR1    4459C
4473  ERR2    4499B
4480  ERR21    4475   4478B
4479  ERR21A
4488  ERR3     4463   4474   4481   4483B  4486C  4505   4510B
4487  ERR31    4468   4479
4493  ERR34    4489   4491B  4501
4499  ERR4     4496B
4502  ERR5     4472B
5039  ERR6     5007   5008   5009   5017   5018   5020
3622  ERRINT   3710
4455  ERROR     457B   607B   612B   618B   626B   647B   663B   667B   673B   689B
               694B   698B   704B   708B   721B   781B   791B   801B   996B  1022B
              1027B  1039B  1140B  1182B  1196B  1258B  1269B  1396B  1418B  1423B
              1462B  1471B  1581B  1621B  1641B  1665B  1669B  1673B  1709B  1730B
              1741B  1757B  1813B  1849B  1874B  1879B  1883B  1933B  1941B  2259B
              2270B  2287B  2377B  2390B  2423B  2525B  2534B  2550B  2758B  2775B
              2858B  2895B  3064B  3196B  3222B  3282B  3293B  3360B  3364B  3379B
              3407B  3422B  3432B  3603B  3662B  3669B  3839B  3878B  3886B  3890B
              3894B  3903B  3963B  3966B  4036B  4093B  4226B  4261B  4289B  4294B
              4306B  4364B
3667  ERUPT    3626
2579  FDTA     2222B  2351B
2594  FDTA1    2595B
2587  FDTALP   2589B
 857  FILL      859B
5889  FLAG     2189C  2198C  2256   2323C  3070C  3073   3110   3112C
4071  FLN       596B   656B   682B   747B   828B  1012B  1055B  1216B  1297B  1360B
              1491B  1611B  1654B  1907B  2204B  2341B  2616B  2798B  2898B  2993B
              3148B  3487B  3560B  4087B
4074  FLN2     4077B
4080  FLN3     4075B
6179  FREV     6156
5864  FRST      381    393    803    806C
3817  FTBL     3512B  3542B  3547B
5873  FW-REV   1356   6158C
4085  GENITZ    590B   652B   735B   823B  1010B  1052B  1213B  1352B  1495B  1616B
              1779B  1905B  2220B  2349B  2606B  2796B  2992B  3150B  3482B  6148B
3130  GNBPRG   3012
1088  GRB1     1082B
5929  HRTZ      396C   402   4389   4396
1631  ICON1    1628B
1634  ICON2    1626C
1611  ICON3    1643B
3052  IDLP     3054B
5953  IDMSG     460
3871  IHD1     3780  3905B  5993   6001   6009   6017   6025   6033   6041   6049
               6057   6065   6073   6081   6089   6097   6105   6113
3886  IHDG1    3884B
3891  IHDG2    3886B
3895  IHDG3    3892B  3899
3904  IHDG4    3896B  3901B
3879  IHDG5    3876B
1965  ILC1     1962B
1959  ILCIST   1189B  1937B  2266B  2386B  3193B  3219B  3279B  3357B
6139  ILV      3007C  3082C  3255C  3319C  3515   3528   3756   3835   3891   4015
               4030   4049
5868  IMASK     438C   470   3508
5906  IMGA      593C   605    661    671   4158C  4181
3911  IMON     3617   3687   3921B
3918  IMON2    3913B
3920  IMON4
4301  INBYTE   1136B  1256B  1967B  1971B  2282B  2418B
1606  INCONT    549B
3690  INLLST   3690   3704
2992  INLOOP   3126B
6131  INMB     3085C  3322C  3545   3761   3837   3883   3918C
1572  INS1     1566C
1498  INS2     1496B
1491  INSLUP   1564B
2988  INST1     556B
```

```
3071  INST2    3120B
3125  INST3    3111B
3077  INST4    3074B
4533  INSTR1   1508    1509
4817  INSTR2   1509    1515    1516
4940  INSTR3   1516    1522    1523
5047  INSTR4   1523    1528    1529
5054  INSTR5   1529    1534    1535
5061  INSTR6   1535    1540    1541
1487  INSTT     548B
3682  INTBL    3702
3701  INIIT4    589B    2995B   3071B   3253B   3317B   3483B
3176  INTLCT   3000
5926  INTM     4352C   4367
3710  INTNX    3705B
4221  INXT      787B    1035B   1864B   1865B   1928B   2254B   2368B   2745B   3061B   3960B
                4354B
 609  IOT1      606B
 627  IOT2      625B
 622  IOT3      617B    619B
5900  IRG      1393C   1459C
5989  IS1      3771    3772    4046
3617  IS4S     3686
1040  IS64K    1036B
3613  ISAV4    3732
6135  IST      3004C   3079C   3550    3752    3900
1311  ITEST    1237    1238
1313  ITEST1   1236
3703  ITM1     3706B
4010  ITM11    4016B   4020B   4025B
4022  ITM11A   4017B
4023  ITM12    4024B
3713  ITM2     3714B
3865  ITM7     3823B
1305  IZL      1790
1303  IZLCT    1245
1651  IZTST     550B
1846  IZTST1   1857B
1834  IZTST2   1831B
1856  IZTST3   1853B
1837  IZTST4   1829C
1654  IZTST6   1815B
1850  IZTST7   1855B
 620  JMPEXT    597B
 410  JP1       407B
 425  JP2       417B
3652  L63HDL   3644    3653B   3685
3640  L63SV1   3734
3644  L63SV2   3041C   3097C   3260C   3334C   3684
3646  L63SV3   3044C   3100C   3263C   3337C   3582C
6184  LAF8     6181
5978  LAF98
5976  LAF99
3671  LAST     3660    3667
2556  LCT1     2082    2096    2138    2151    2338C
2555  LCT1A    2110    2124
2564  LCT2     1993    2006    2048    2061    2201C
2563  LCT2A    2020    2033
3862  LCT3     3863B
3854  LCT4     3864B
3860  LCT5     3856B
1163  LCTA     1158C
1055  LCTB     1201B
1145  LCTC     1148B
1160  LCTE     1156B
1068  LCTGRB   1063B   1657B   1689B   1910B
1050  LCTOB     545B
1588  LCTTAB   1557
1163  LCTW     1178B
1175  LCTX     1183B
1117  LCTZ     1064B
3398  LEVRCH   3028B   3252B   3316B
5865  LOOP      532C
5860  LRCB
6116  LS1      5887
1728  LUP      1732B
1731  LUP1     1729B
3049  LV631A   3040    3096
3271  LV631B   3259
3513  LVSTR    3470C   3505    3519C
5913  MASK      496     742C    794     822C    989     1281C   2627C   2658C   2665C   2768
                2803C   3228C   3234    3237C   3249    3537C
5914  MASK1     826C   1500C   1578    3751C   3752    3756    3760C   3761    3770C   3771
 998  MBM       991B
4280  MCCB      765B    772B   1690B   1914B   2477B   2729B   2763B   2828B   2833B   2838B
                2843B   2861B   3022B   3093B   3114B   3493B   3945B   3991B
 820  MEMT      543B
 828  MEMTA     833B    844B
 834  MEMTB     831B
6180  MESG      390
5952  MESG1     403
 526  MESG2     495
6187  MESG5     401
3775  MISA     3788B
3793  MISB
5872  MLCADD    485C   3875
      MLCS1       1    6193
1007  MNKT      544B
 955  MTM10     927B
 866  MTM6      863B
4875  NEXT     4870    4871    4872
4900  NEXT1    4892    4893    4895
4911  NEXT2    4906    4907    4908
4931  NEXT3    4926    4927    4929
3079  NEXT4    3077B
3495  NEXT5    3497B
 682  NL1       724B
 656  NLINE     675B
 428  NODEVF    413
 803  NOER1     796B
3273  NOER20   3274B
1198  NOER5    1188B
2778  NOERC    2770B
```

```
 525  NOGO      522
1562  NOINF     1580B
1097  NONO      1080
 540  NOP
 578  NOTTY
1407  NP4       1408B
1420  NP5       1417B
1428  NP6       1422B
1432  NP7       1433B
 735  NXCB      810B
 741  NXCB1
 772  NXCB2
 776  NXCB3
 596  NXCH      645B
2616  NXCHC     2620B   2672B
2621  NXCHC1    2616B
2627  NXCHC2    2625B
2614  NXCHC4    2617B
1071  NXLCT     1076B
1080  NXLCT1    1084B
1063  NXLCTA    1058B
2204  NXRCV     2210B   2290B
 603  NXRG      637B
2374  NXT1      2372B
2424  NXT2      2422B
 659  NXTCCB    670B
2341  NXWRT     2347B   2425B
4513  OPT       4466    4469
2792  OTPT      555B
2898  OTPT1     2902B
2798  OTPT2     2802B
2885  OTPT3     2870C
2803  OTPT4     2800B
2824  OTPT5     2900B
2860  OTPT7     2857B
3417  OUTLCT    1135B   1255B   1966B   1970B   2281B   2417B   3191B   3214B   3241B   3250B
                3267B   3329B   3347B   3371B
 580  PASMSG    568
5866  PASS      386C    569C
4327  PKR1      4333B
4323  PKRIN     1151B
5960  PLB       588B    733B    821B    1008B   1051B   1211B   1350B   1488B   1609B   1652B
                1904B   1987B   2000B   2027B   2042B   2055B   2076B   2090B   2104B   2118B
                2132B   2145B   2605B   2793B   2989B   3146B   3467B
5969  PLB1      5966B
5973  PLB2      5970
5974  PLB3      5969C
6147  PREV      515B    805B
2897  PRFK1     2892B
3691  PRIEND    3912
3472  PRILP     3593B
6142  PRIST     3689    3691    3692    3765
1324  PROG1     1376
1328  PROG2     1370
1331  PROG3     1359C
5898  PRTFLG    493C    503C    5965
 568  PSPT      562B    564B
3450  PVLU1     3446C
 931  Q1        920C    926
 932  Q2        916C    950
 942  Q3        921C
 943  Q4        917C
 962  Q5        913C    955
 972  Q6        914C
5896  QFLG      494C    510C    563     570     631
5255  R56       5210    5211    5215    5221    5222    5223    5233    5234    5238    5244
                5245    5246
5202  R57       5095    5096    5098
5213  R58       5103    5104    5106
5225  R59       5111    5112    5114
5236  R60       5119    5120    5122
5248  R61       5127    5128    5130
5155  R67       5150    5151    5152
5192  R68       5158    5159    5161
5195  R69       5172    5173    5174
5170  R70       5146    5147    5148
5181  R71       5176    5177    5178
3495  RAD       3475C   3500C   3501C
5904  RAMAD     850C    992
5905  RAMAD2
5901  RANGE     744C    763     771     1016C   1678C   1679
5902  RANGE1    2613C   2708    2768
5903  RANGEW    746C    794     849C    855     883C    923C    957C    983     989     1283C
                1570    1576    1773C
1141  RBYTE1    1133B   1139B
1130  RBYTES    1141B
1986  RC0       552B
1999  RC1       1989B
2236  RC11      2214C
2013  RC2       2002B
2026  RC3       2016B
2041  RC4       2029B
2054  RC5       2044B
 679  RCCBS     657B
2223  RCDATA    2215C
5306  RCDT
5304  RCDTC1    2246    2247    2360
2292  RCEND     2207B
2472  RCLCT     2459C
5531  RCRT1     2247    2634    2635
2441  RCV       2243B
2225  RCV1      2171C
2248  RCV10     2183C
2288  RCV11     2286B
2211  RCV12     2206B
2199  RCV13     2191B   2196B
2197  RCV14     2192B
2464  RCV1M     2462B
2237  RCV2      2172C
2465  RCV2M     2454C
2249  RCV3      2173C
2478  RCV3M     2455C
2275  RCV4      2174C
```

```
2479  RCV4M    2476C
2224  RCV5     2167C   2194C
2235  RCV6     2186C
2245  RCV7     2177C   2190    2193
2276  RCV8     2186C   2195C
2250  RCV9     2180C   2261
2166  RCVTST   1988B   2001B   2015B   2028B   2043B   2056B
2466  RCWR     2457C
2467  RCWR1    2168C   2443C   2506C
3982  RDATA     874B    896B    940B    970B   1160B   1275B   1569B   1631B   1719B   1834B
                2400B   2882B   3447B   6149B
3992  RDTA1    3984C
3993  RDTA2    3987C
 811  READ      761C    776
 386  RESTRT    377     409B    507B   6175B
5897  RFLG      520C    565     572
 647  RGE       628B
 587  RGNT      541B
 631  RGNT1     648B
 636  RGNT2     632B
 637  RGNT3     634B
5533  RMCRT     5672    5673    5674
5594  RMCRT1    5607    5608    5609
5668  RMCRT2    5663    5664    5665
5580  RMCRT3    5543    5544    5545
5633  RMCRT4    5559    5560    5562    5598    5599    5601    5616    5617    5619
5551  RMCRT5    5576    5577    5578
5612  RMCRT7    5590    5591    5592    5629    5630    5631
5651  RMCRT8    5646    5647    5648
5922  RMINST     773    5908
4566  RMUP
5666  RMUPT
5706  RMUPT1    5735    5736    5737
5718  RMUPT2    5713    5714    5716
5696  RMUPTA    5753    5754    5756
3047  RNFC      3036C   3053
 685  RNXR       712B
5684  ROD1      2635    2808    2809
1907  RPS10     1951B
3438  RPVLU     1265B   1390B   1414B   1455B   1609B   2538B   3289B   3375B
 393  RST        386B    426B
 403  RST1       394B    399B
3505  RT1       3488B
3566  RT10      3575B
3544  RT2       3535C
3549  RT3       3536C
3525  RT5       3518B
3507  RT6       3521B
3528  RT7       3533B
3520  RT7A      3509B
3533  RT8       3531B
5979  RTB        462C    463     749C    750C    751C    752C    766     794     875     897
                 941     971     983     989    1161    1175    1276    1679    1720    1728
                1835    1846    1891    1915    2219    2244    2401    2519    2592    3023
                3094    3115    6150    6157    6165    6170
4440  RTC       3042B   3098B   3261B   3335B   3579B
3678  RTCFLG    3676C   3737C
3676  RTCHDL    3635    3683
5923  RTCHN     3401C
3631  RTCSAV    3730
3635  RTCSV1    3682
5081  RTD       5163    5164    5165    5185    5186    5187    5289    5290    5292
5084  RTD1      5077    5078    5079    5281    5282    5285
5136  RTD2      5277    5278    5280
 512  RUT3       509B
 518  RUT3A      513B
 522  RUT3B      519B
 495  RUTBG      504B    511B    516B    523B    571B    644B   4100B   5963B
 506  RUT1       502B
4049  SAF4
4050  SAF5
4514  SAFF      4504
5916  SAV1      3701C   3796    3817C   3827    3832C   3842    4006C   4011    4071C   4080
                4254C   4263    4280C   4296    4323C   4334    4343C   4371    4388C   4395C
                4432
5917  SAV2      1959C   1974    3850C   3857    3936C   3967    3982C
5918  SAV3      2441C   2488    2502C   2579C   2596    3598C   3604    4085C   4101
5919  SAV4      2166C   2292    2307C   2427    4150C   4184
5921  SAV5      4455C   4473    4485    4502
5920  SAV6      4221C   4228
5909  SBCCB      740C    794
5910  SBCCP
5908  SBCK       737
 652  SCCBTS     620B
5894  SCFLG     3324C   3739C   3895
3313  SCIT1     3310
3299  SCIT1A    3236B
3316  SCIT2     3311B   3349
3344  SCIT3     3333    3348
3394  SCNPRG    3156
3172  SCTEST    3169    3233    3303
3188  SCTST1    3170B
3412  SCTSVA    3272
3936  SDATA      866B    886B    929B    960B   1228B   1236B   1369B   1375B   1507B   1514B
                1521B   1527B   1533B   1539B   1548B   1781B   2233B   2464B   2633B   2807B
                3011B   3155B
5977  SDB        854     867     887     930     961     989    1073C   1083C   1089C   1091C
               1092C   1117C   1119C   1121C   1123    1127    1144    1152    1153    1173
                1221    1229    1691C   1692C   1693C   1694C   1695C   1696C   1698    1912
                2234    2357    2518    2580    2582C   2610    2642    2696    2768    2783
                2817    2829    2834    2839    2844    2862    2875    3472   6147C   6176
5924  SEC        533C    560     601     602C   4381C
 983  SET        852B    893B    937B    967B   1774B
4047  SET1      4056B
4053  SET2      4051B
1889  SET4      1701B   1745B   1760B   1793B
1890  SETA      1895B
3850  SETLCT    1122B   1244B   1382B   1400B   1428B   1446B   1556B   1697B   1751B   1789B
                1911B   2471B   2705B   2824B   2999B   3118B   3168B   3232B   3271B   3302B
                3309B
3675  SETNEW    3677B
4505  SETNR     4477B
4029  SETV      3558B
```

```
4038  SETV1     4031B
4030  SETV2     4040B
4045  SETVEC    3032B   3086B   3320B   3554B
3122  SFLG      3119
1213  SIT1
1285  SIT10
1275  SIT11     1272B
6165  SIT12
1252  SIT2
1228  SIT4      1225B
1259  SIT5      1257B
1254  SIT6      1260B
1271  SIT7      1268B
1224  SIT9      1299B
1210  SITEST     546B
3466  SITST      558B
3823  SLVLP     3826B
5871  SPACE     4462    4507
4511  SPACE1    4492
3947  SPRG2     3941C
3954  SPRG3     3998B
3946  SPRG5     3939C
3959  SPRG7     3954B
3961  SPRG8     3958B
4535  ST        4826    4827    4829
5928  STAT      4227C   4429C
1903  STIORC     551B
3692  STKPTR    3688    3911    3916C
 532  STLOOP     382B    500B
 377  STOP       578C    579B
 381  STRT       383     576B    6193
1467  STRT10    1250B   1386B   1405B   1430B   1558B   1795B   1803B   2483B   2742B   2848B
3722  SVEC      3723B
4512  SWT       4465C   4482    4490    4506C
4014  TALL      4010B
1602  TALU      1549
5891  TEMP      1632    1638    1667    1872    3448    3453    3455C   4153C   4161C   4175
                6161C   6162    6167C   6168
5892  TEMP1      861C    979    1494C   3207C   3247    4154C
5893  TEMPST    3213C   3328
2728  TESB      2737B
2686  TESC      2648B   2654B   2662B   2669B   2693
2689  TESCA     2698B
2699  TESCC     2695B
2759  TESD      2754B
2753  TESE      2746B
2742  TESF      2755
2745  TESG      2766B
2747  TESH      2756B
2766  TESJ      2760B
 847  TESM       836B    838B    840B    842B
 979  TESM1      951B
4343  TEST      2852B   3955B
4355  TESTZ     4363B
4366  TESTZ1    4356B   4358B
5888  TIMEIN
6121  TOPSAV    3727    3794C   3795C
5890  TPK        446C    464     468C    469     496     497     757C    770     792     862C
                 864C    865C    873C    885C    895C    925C    939C    959C    969C   1224C
                1249    1271    1295    1365C   1493C   1582    1627C   1637    1682    1712C
                1716    1733    1830C   1840    1968C   1973    2461C   2470    2540    3009C
                3016    3153C   3161    3439C   3456    4443C   4445
2075  TRAN1      553B
2089  TRAN2     2078B
2103  TRAN3     2092B
2117  TRAN4     2106B
2131  TRAN5     2120B
2144  TRAN6     2134B
5075  TRANM1    1541    2359    2360
4006  TRNON     3049B   3566B   3570B
3660  TRPERR
5907  TSAZ      3746
1203  TSTCHN    1057C   1155    1198
1317  TSVLU     1252
5925  TTOT       534C   4375    4376C   4380C
4375  UPTM      4369B   4401B   4444B
4382  UPTM1     4379B
5764  VALTST    1782    1783    2809
6173  WFW
2502  WRT       2356B
2453  WRT1      2507B
3348  WRUPT     3350B
5481  XM56      5435    5436    5437    5447    5448    5452    5458    5459    5460    5470
                5471    5475
5427  XM57      5326    5327    5329
5439  XM58      5334    5335    5337
5450  XM59      5342    5343    5345
5462  XM60      5350    5351    5353
5473  XM61      5358    5359    5361
5386  XM67      5381    5382    5383
5417  XM68      5389    5390    5394
5421  XM69      5405    5406    5407
5403  XM70      5377    5378    5379
5392  XM71      5409    5410    5411
5312  XMD       5396    5397    5398    5513    5514    5515
5315  XMD1      5308    5309    5310    5506    5507    5508
5367  XMD2      5502    5503    5504
5074  XMDT
2307  XTST      2077B   2091B   2105B   2119B   2133B   2146B
2348  XTST1     2345B
2322  XTST2     2320B
2333  XTST3     2331B
 361  ZERO       376    5976    5978    5980
      ZHCOMM     367
      ZHIAFB     364    3740C   3741C   3742C   3744C
      ZHISAZ     364    3711    3721    3726    4052C   4399C
      ZHNTSA     362     412C   3725C
      ZHPFR      367    3719    4344    4398
      ZHRTCC     365    4351C   4357    4368    4400C   4423    4426    4442C
      ZHRTCI     366    4346C   4403C   4441C
      ZHRTCL     366    4348C   4405C   4447C
      ZHTH15     362     414C
      ZHTH17     363
```

```
          ZHTSA      363      411     3724     3747C
          ZHWDTC     365
          ZV$BRF     368      643     4099     5962
          ZV$BRK     368      642B    4098B    5961B
          ZV$C       794B     989B    2768B
          ZV$CO      802B     997B    2777B
          ZV$ER      425B    4460B
          ZV$F       436B     983B    1221B    1679B    2219B    2708B    3752B    3756B    3761B    3765B
                    3771B
          ZV$HM     1017B
          ZV$HR      362     2692
          ZV$IA      368      496B
          ZV$ID      402B
          ZV$IH      404B
          ZV$PCH     408B
          ZV$QC      401B     403B     495B
          ZV$RD      363      390B    1017B
          ZV$I       465B    4492B    6163B
          ZV$TC      460B     522B     568B    5970B    6156B    6173B
          ZV$TD      463B    6162B    6168B
          ZV$TH      469B    4487B
          ZV$THZ    4488B
          ZV$TTY     368      575
 693 LABELS
4114 REFERENCES
6193 RECORDS
   0 U FLAGS
   0 M FLAGS
  44 N FLAGS
6 CROSS REF VERSION L - 24 SEPT, 1976
RS LINKER VERSION 5.00 08/16/78  1115.0 EDT FRI
LINK MAP FOR MLCS1
 START          0100
 LOW            0000
 HIGH           27FF
 CURRENT        1D5F
*LOC DEFS
  ZHCOMM        0000
 *MLCS1         0000  REV F
  ZHPFR         0000
  ZHTSA         0002
  ZHNTSA        0010
  ZHKTC1        0014
  ZHRTCC        0015
  ZHRTCL        0016
  ZHWDTC        0017
  ZHMERC        001F
  ZH1AFD        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-UP        0073
  ZHTH12        0074
  ZHTH11        0075
  ZHTH10        0076
  ZHTH9         0077
  ZHTH8         0078
  ZHTH7         0079
  ZHTH6         007A
  ZHOVFL        007A
  ZHTH5         007B
  ZHOP-N        007B
  ZHTH4         007C
  ZHTH3         007D
  ZHSC-N        007D
  ZHTH2         007E
  ZHTRC         007E
  ZHTH1         007F
  ZHMCL         007F
  ZHISAZ        0080
  ZH1VBS        0080
  ZHTVBS        0080
 *ZV$TH         1492
  ZV$TD         14C7
  ZV$TH         1492
  ZV$THZ        14BA
 *ZV$IH         14E2
  ZV$ID         14E7
  ZV$IH         14E2
  ZV$IAD        14EC
  ZV$--2        1504
  ZV$--3        1516
 *ZV$PCH        157B
  ZV$PCH        157B
 *ZV$ER         167D  REV. 5.0
  ZV$ER         167D
  ZV$IA         16A9
  ZV$--0        1690
 *ZV$F          16ED
  ZV$F          16ED
 *ZV$T          16FB  REV. 5.0
  ZV$QC         1718
  ZV$TC         1704
  ZV$T          16FB
  ZV$0          170D
```

```
*ZV$IA          172C  REV. 7
 ZV$IA          172F
 ZV$ABF         17E0
 ZV$ARG         17DE
 ZV$--1         179B
 ZV$IAV         172D
*ZV$BRK         17EB
 ZV$BRK         17EB
*ZV$C           1805  REV. 5
 ZV$C           1805
 ZV$CO          1828
*ZV$GP          1839
 ZV$GP          1839
 ZV$--4         1859
*ZV$HA          1865
 ZV$HA          1865
 ZV$HZ          186F
 ZV$HS          186A
*ZV$HD          189E
 ZV$HD          189E
*ZV$KD          18D0  REV. 7
 ZV$KD          18D0
 ZV$ITY         18E3
 ZV$BKF         18F8
 ZV$HM          1946
 ZV$HK          18FF
 ZV$--5         1902
 ZV$SV1         1AA5
 ZV$SV3         1AC5
 ZV$AF          18E1
 ZV$SV2         1AB5
 ZV$UTP         1977
 ZV$TID         18E2
 ZV$CF2         18EC
 ZV$TK          18E8
 ZV$RAR         18E9
 ZV$STI         18ED
 ZV$RCC         18EE
 ZV$BUD         18E4
 ZV$OLB         18F0
 ZV$RCB         18F1
 ZV$NSR         18F5
 ZV$STR         18F3
 ZV$BKS         18F7
 ZV$IZ          190A
 ZV$LR          18FC
 ZV$DAT         18DF
 ZV$HRU         18F9
 ZV$HRL         18FA
 ZV$LRU         18FB
 ZV$LRL         18FC
 ZV$HBD         18FD
 ZV$CF1         18EB
 ZV$RMD         18E0
 ZV$MCP         18FE
 HIBAUD         18FD
 ZV$RAW         18EA
 ZV$RDT         1801
 ZV$CTL         18E7
 ZV$B1          1A22
 ZV$TSI         1B57
 ZV$MDC         1B2B
 ZV$R99         1D29
 ZV$ISA         1905 .
 ZV$UIH         1900
 ZV$ZRO         1984
 ZV$BSM         1986
 ZV$CPU         18E6
 ZV$R50         1964
 ZV$R60         196F
 ZV$R1          1C66
 ZV$ALL         18E5
*MLCHPG         1D2E  T+V
 MLCHPG         1D2E
 ENDCHP         1D5F
*UNLINK MODULE(S)
 ZV$GC
 ZV$ID
 ZV$TC
 ZV$TD
 ZV$CO
 ZV$TH
```