```
 TITLE DCMS3, 'REV A'
*      BBHCLA TEST
*      PART NO.
* DCMX3      60134626-001
* DCMS3      60134627-001
* DCML3      60134628-001
*
* DESCRIPTION
* -----------
*      THIS T & V PROGRAM VERIFIES PROPER OPERATION OF THE LEVEL-6 MLCP
*      BROADBAND HDLC COMMUNICATION LINE ADAPTER (BHCLA).  IT PROVIDES
*      A FIRST LEVEL OF DIAGNOSIS WHEN FAILURES ARE DETECTED, AND MAKES
*      FACILITIES AVAILABLE TO SUPPORT EXTENSIVE PROBLEM INVESTIGATIONS.
*
*      THE SUBSYSTEM OPTIONS SUPPORTED BY THIS PROGRAM ARE:
*
*      MLC 9101  MLCP CONTROLLER
*      DCM 9112  BH4DLD PAC - BROADBAND HDLC, CURRENT MODE.
*      DCM 9113  BH4DLE PAC - BROADBAND HDLC, BALANCED VOLTAGE.
*
* REVISION HISTORY
* -------- -------
*      001     JUNE  1978     DCMS3/DCML3             ORIGINAL RELEASE
*
*      THIS DOCUMENT AND THE INFORMATION CONTAINED THEREIN IS CONFIDENTIAL AND
*  PROPRIETARY TO AND THE EXCLUSIVE PROPERTY OF HONEYWELL INFORMATION SYSTEMS
*  INC.   IT  IS MADE AVAILABLE ONLY  TO  HONEYWELL AUTHORIZED RECIPIENTS FOR
*  THEIR USE SOLELY  IN  THE MAINTENANCE AND OPERATION OF HONEYWELL PRODUCTS.
*  THIS DOCUMENT AND INFORMATION MUST BE MAINTAINED  IN STRICTEST CONFIDENCE;
*  IT MUST NOT BE REPRODUCED IN  WHOLE OR IN PART;   AND IT SHALL NOT BE DIS-
*  CLOSED TO ANY OTHER PARTY WITHOUT THE PRIOR WRITTEN CONSENT OF HONEYWELL.
*****************************************************************************
* PROGRAM PREPARATION:
* ------- -----------
*      THE ROOT SOURCE OF THIS PROGRAM, AFTER THE ADDITION OF THE APPROPRIATE
*      TITLE AND END STATEMENTS, WAS PROCESSED BY THE HOST RESIDENT ASSEMBLER
*      TO CREATE EITHER SHORT OR LONG ADDRESS FORM ( SAF OR LAF ) OBJECT TEXT
*      AND LISTING.   THE  OBJECT TEXT  WAS  FURTHER PROCESSED  BY  THE  HOST
*      RESIDENT LINKER USING THE APPROPRIATE CONSOLE ZV$LIB LIBRARY TO CREATE
*      A PUNCH SEGMENT CONTAINING AN EXECUTABLE MODULE.  THE ASSEMBLY LISTING
*      WAS AUGMENTED WITH CROSS REFERENCE DATA,  PLUS  THE LOAD MAP FROM  THE
*      LINKER TO CREATE A LIST SEGMENT.
*                ROOT              SAF              LAF
*                ----              ---              ---
*      NAME      DCMX3             DCMS3            DCML3
*      DOCUMENT 60134626-001     60134627-001     60134628-001
*
* DISTRIBUTION
* ------------
*      THE ELEMENTARY ITEMS SUBMITTED TO THE T & V PROGRAM DISTRIBUTION CENTER
*      WERE THE EXECUTABLE LINKED IMAGES, ON DISKETTE, OF DCMS3 AND DCML3, AND
*      MAGNETIC TAPE IMAGES OF THE AUGMENTED LISTINGS.
*
*      REPRODUCTIONS OF THE EXECUTABLE LINKED IMAGES MAY BE  AS DUPLICATE CARD
*      DECKS OR  AS A MEMBER OF A MULTIPLE MEMBER FILE.   IN THE MOST FREQUENT
*      CASE, IT WILL BE FOUND  AS  MEMBER "SP" (SAF) OR "LP" (LAF) WITHIN FILE
*      "PROGFILE" OF A DISKETTE VOLUME ENTITLED "DIAGS".
*
*      DISTRIBUTION OF THE LISTINGS,  WHICH SHOULD BE AVAILABLE IF ANY COMPLEX
*      MAINTENANCE OR REPAIR IS TO BE PERFORMED, IS NORMALLY AS A PRINTED COPY.
*
* ROUTINE DEMONSTRATION
* ------- -------------
*         A MINIMUM SATISFACTORY TEST FOR NORMAL OPERATION MAY BE OBTAINED  BY
*         RUNNING ONE PASS OF "A" MODE AND ONE PASS OF "C" MODE AT 56
*         K BAUD OR LESS FOR EACH BBHCLA. OPERATION AT 72 KB RUNS ONLY A
*         SUBSET OF THE PROGRAM.  ONLY SINGLE FRAME TESTS WITH CHAR. SIZE
*         OF 8 BITS ARE RUN AT GREATER THAN 56 K BAUD.
*
*
* STORAGE
* -------
*      THIS PROGRAM REQUIRES AT LEAST 16 K WORDS OF MAIN MEMORY.
*
* OPERATION
* ---------
*      LOAD AND START (OR RESTART) THE PROGRAM. THE PROGRAM IDENTIFICATION WILL
*      BE DISPLAYED ON THE CONSOLE.  THE INITIAL START WILL ALSO DISPLAY:
*
*        THE ZV$LIB REVISION NUMBER
*        THE ADDRESS FORM (SAF OR LAF)
*        I/O  EQUIPMENT DETECTED IN THE SYSTEM
*        MEMORY SIZE
*
*      THIS DISPLAY MUST BE VERIFIED BY THE OPERATOR.   THIS DISPLAY IS OMITTED
*      ON RESTARTS.
*
*      THE CONSOLE SEARCH RULES ARE:  FIND THE CONSOLE WITH THE LOWEST  CHANNEL
*      NUMBER CONNECTED THRU AN  MDC  CONTROLLER.  IF THERE IS NO CONSOLE ON AN
*      MDC, THEN SEARCH FOR A TERMINAL WITH THE HIGHEST CHANNEL NUMBER ASSIGNED
*      TO AN ACLA ADAPTER ON AN MLC CONTROLLER.  IF NO ASYNC ADAPTER  IS FOUND,
*      THEN GO TO THE FULL CONTROL PANEL.
*
*      THERE ARE THREE CONSOLE CHANNEL OPTIONS DETERMINED BY THE VALUE  OF  LO-
*      CATION "ZV$TTY".
*
*      IF ZV$TTY EQUALS (0000), SEARCH FOR A CONSOLE.
*      IF ZV$TTY EQUALS (FFFF), ASSUME THERE IS NO CONSOLE.
*      IF ZV$TTY EQUALS NEITHER (0000), NOR (FFFF), THEN IT IS THE CONSOLE CHAN-
*      NEL NUMBER.    NOTE:  DEFAULT IS TO SEARCH FOR A CONSOLE.
*
*      ALL CONSOLE I/O IS EVEN PARITY.  IF CONSOLE IS ON MLC, IT MUST BE ASYNC
*      AND THE BAUD RATE SET AT 1200 TO MATCH THE PROGRAM SUPPLIED RATE.  IF IT
*      IS NECESSARY TO CHANGE THE PROGRAM BAUD RATE,   THEN THE  NEW  BAUD RATE
*      CODE SHOULD BE PUT INTO LOCATION "ZV$BUD" IN HEX. THE TERMINAL BAUD RATE
*      MUST BE  SET  TO MATCH THIS NEW BAUD RATE.  THE CORRECT HEX VALUE MAY BE
*      OBTAINED FROM THE FOLLOWING TABLE.
*
*-----------------------------------------*
*                                         *
*         BAUD  RATE  TABLE               *
*                                         *
*-----------------------------------------*
```

```
* ACLA I.D.           (2118)(2110)         (2108)
* BAUD-RATE
*    50                   0                   1
*    75                   1                   2
*   110                   2                   3
*   134                   3                   4
*   150                   4                   5
*   200                   5                  ---
*   300                   6                   6
*   600                   7                   7
*   900                  ---                  8
*  1050                   8                  ---
*  1200                   9                   9
*  1800                  10 (A)              10 (A)
*  2000                  11 (B)              ---
*  2400                  12 (C)              11 (B)
*  3600                  ---                 12 (C)
*  4800                  13 (D)              13 (D)
*  7200                  ---                 14 (E)
*  9600                  14 (E)              15 (F)
* 19200                  15 (F)              ---
*
*      TO MAKE ANY OF THE ABOVE CHANGES,  LOAD AND HALT THE PROGRAM BEFORE EX-
*      ECUTION.  INSERT CHANGE THEN EXECUTE.  MEMORY LOCATIONS OF "ZV$TTY" AND
*      "ZV$BUD" MAY BE FOUND IN MAP AT END OF LISTING.
*      CONSULT LEVEL-6 T&V MANUAL "AW94" FOR DETAILS ON HOW TO LOAD THE TESTS.
*
*      THE FOLLOWING IS A TYPICAL RESULT OF LOADING  AND  STARTING TO  RUN THE
*      PROGRAM.
*
*      BHCLA TEST  DCMS3  DEC 13, 1977  REV A
*      ZV$LIB REV. 6.0
*      ZV$AF= 1  <2>
*      WDT
*      CHAN  DEVC  ID
*      0400  DSKT  2010
*      0480  DSKT  2010
*      0580  CDR   2008
*      1200  DISC  2330
*      1280  DISC  2330
*      1300  LPT   2000
*      1380  CONS  2019
*      MEMORY LOW  00002B2D
*      MEMORY HIGH 00003FFF  16K
*
*           FOR 6/3X OPERATION THE PROGRAM WILL NEXT DISPLAY:
*
*                  PWR FREQ (HZ)?:
*
* THE OPERATOR MUST RESPOND WITH THE FREQUENCY OF THE CLOCK USED
* AS A SOURCE FOR THE REAL TIME CLOCK (EG. 50 HZ OR 60 HZ).
* THE PROGRAM USES 60 HZ FOR 6/40 OPERATION.
*
* THE PROGRAM NEXT DISPLAYS;
*
*                 BHCLA CHANNEL(S)?:
*
* THE OPERATOR MUST RESPOND WITH THE  ADDRESSES OF BHCLA'S
* TO BE TESTED. DELIMIT ADDRESSES WITH COMMA'S AND TERMINATE
* WITH A CARRIAGE RETURN.
* BHCLA ADDRESSES WILL NORMALLY BE DISPLAYED AS ITEMS IN THE
* CONFIGURATION PRINTOUT.
*
*   NOTE; IF IT IS DESIRED TO TEST BHCLA'S  ON MORE
*         THAN ONE MLCP MOTHER BOARD, THE PROGRAM MUST BE
*         RESTARTED AFTER TESTING THOSE ON THE FIRST
*         MOTHER BOARD. THIS IS ACCOMPLISHED BY RESTARTING AT
*         HEX 105 IF THE FULL CONTROL PANEL IS PRESENT OR
*         RELOADING THE PROGRAM IF ONLY THE BASIC PANEL IS
*         PRESENT.
*
* THE PROGRAM NEXT DISPLAYS;
*
*           LINE XX   ID = 21F6   FOR BH4DLD
*
*   OR
*
*           LINE XX   ID = 21F7 FOR BH4DLE
*
*   WHERE XX IS THE LINE NUMBER (1 OR 4)
*
* THE OPERATOR SHOULD VERIFY THIS PRINTOUT CORRECTLY MATCHES
* HIS SYSTEM CONFIGURATION.
*
* FOR USERS WITHOUT A CONSOLE, LINE ADAPTER ID FOR  LINE 0 IS
* STORED IN LOC ATLT AND FOR LINE 1 IN ATLT+1.
*
*
* THE PROGRAM NEXT DISPLAYS;
*
*         NEXT?:
*
* THE OPERATOR ENTERS;
*
*     "A"    RUN TEST WITH LINE ADAPTER INTERNAL LOOP.
*            RUNS FOR APPROX. 3 MIN/PASS.
*
*     "Q"    QUICK FLAG. RUN ONLY ONE PASS OF NEXT TEST SPECIFIED.
*
*     "P"    PRINT FLAG. PRINT TEST LABLES FOR NEXT TEST SPECIFIED.
*
*     "I"    INITIALIZE - CLEAR Q,P FLAGS. RESTART PROGRAM.
**
*
*     "E"    EXTERNAL MODE. - ASSUMES DATA LOOP BACK AT MODEM OR
*            OTHER EXTERNAL WRAP. THE DATA CLOCK MUST BE SUPPLIED
*            EXTERNALLY.
*
* OR
*
*     "C" RUN LINES WITH CONNECTOR LOOP.
*
*
```

```
*
* THE PROGRAM THEN DISPLAYS (1ST PASS ONLY);
*
*        MLCP FW REV AA
*
* AND THEN
*
*        BHCLA  LINE     0  FW REV BB
*        DATA SET STATUS =QQQQ
*
*             AND/OR
*
*        BHCLA  LINE     4  FW REV CC
*        DATA SET STATUS = PPPP
*
* WHERE AA, BB, AND CC ARE FIRMWARE REV NUMBERS AND PPPP AND QQQQ
* ARE DATA SET STATUS VALUES READ AFTER AN MLCP  INITIALIZE.
*
*
* THE MLCP REV NUMBER IS STORED IN LOC "MLC-FR".
* BHCLA FIRMWARE REV NUMBERS ARE STORED IN LOCATIONS
* HCFW-1 AND HCFW-2 FOR LINE ADAPTERS 1 AND 2
* RESPECTFULY. THESE ARE FOR EXAMINATION, IF
* DESIRED BY USERS WITH NO CONSOLE.
*
*
*
*
* THE PROGRAM WILL RUN A FEW SECONDS PER BHCLA IF THERE ARE NO
* HARDWARE FAULTS. IT WILL THEN DISPLAY;
*
*        LINE 0 SPEED = AAAA BITS/SEC
*
*             AND/OR
*
*        LINE 4 SPEED = BBBB BITS/SEC
*
* THESE VALUES DEVIDED BY DEC "10" ARE STORED IN LOCATIONS
* HCLS-1 AND HCLS-2 FOR USERS WITHOUT A CONSOLE. A HEXIDECIMAL
* TO DECIMAL CONVERSION MUST BE MADE WHEN INTERPRETING THESE
* MEMORY LOCATIONS.
*
* THE TESTED SPEED CORRESPONDS TO THE SETTING OF THE TEST
* CLOCK ON THE MLCP. THIS CAN BE VARIED BY MEANS OF A HEX
* ROTARY SWITCH ON THE MLCP.
*
*
*
* PASS TIME FOR THE PROGRAM IS APPROXOMATELY 2 MIN.  PER BBHCLA
* FOR "A" MODE OR "C" MODE.
*
* AT THE END OF THE PASS TIME THE PROGRAM WILL DISPLAY;
*
*        PASS
*
* AND CONTINUE ON TO THE NEXT PASS. FOR SYSTEMS WITHOUT A  CONSOLE
* THE PROGRAM WILL HALT WITH EO = HEX 100 AT THE END OF EACH PASS.
*
*
*
*   IF THERE  IS NO CONSOLE PRESENT REFER TO THE MANUAL "SERIES 60
*   LEVEL 6 T + V OPERATOR' S MANUAL" , DOC AW94 FOR INSTRUCTIONS
*   ON ENTERING DATA AND INTERPRETING PROGRAM MESSAGES.
*
**********
*
* CONNECTOR LOOP INFORMATION
*
* THE LOOP CONNECTOR MUST BE AS FOLLOWS. PINS REFERENCED
* ARE ON THE OUTPUT CONNECTOR OF THE LINE ADAPTER.
*
*   301/303 CURRENT MODE ADAPTER  - BH4DLD
*
*   PIN  SIGNAL       TIES TO     PIN  SIGNAL
*
*   22    RTS  LR2 BIT 1           26    DSR  LR5 BIT 0
*   22    RTS  LR2 BIT 1           27    CD   LR5 BIT 2
*   22    RTS  LR2 BIT 1           25    CTS  LR5 BIT 1
*   20    DTR  LR2 BIT 0           24    RING  LR5 BIT 3
*   6     TR DATA                  14    RC DATA
*   4     TEST CLK A                2    TR CLK
*   12    TEST CLK B               10    RCV CLK
*
*   V.35 BALANCED LINE ADAPTER (BH4DLE)
*
*   27    RQS  LR2 BIT 1           23    CD   LR5 BIT 2
*   27    RQS  LR2 BIT 1           21    CTS  LR5 BIT 1
*   26    DTR  LR2 BIT 0           22    DSR  LR5 BIT 0
*   26    DTR  LR2 BIT 0           20    RING  LR5 BIT 3
*   19    XMIT DATA (-)            14    RCV DATA (-)
*   18    XMIT DATA (+)            16    RCV DATA (+)
*    6     TEST CLK (+)             2     TR CLK (+)
*    6     TEST CLK (+)            10    RCV CLK (+)
*    8     TEST CLK                 4     TR CLK (-)
*    8     TEST CLK                12    RCV CLK (-)
*
*
*
*   TO OPERATE IN EXTERNAL LOOPBACK:
*
*        1.  DIRECT CONNECT MUST BE SET
*        2.  LR2 BITS 2,3, AND 5 MUST BE RESET
*
*
*
*
* ERROR REPORTS
*
*   ERRORS WILL CAUSE THE PROGRAM TO HALT. AN ERROR MESSAGE WILL
*   BE DISPLAYED IF A CONSOLE IS PRESENT.
*
*   ERROR DISPLAYS ARE AS FOLLOWS;
*
```

```
*              ERR MBXX AT YYYY
*
*              AA   BB   CC   DD   EE   FF   GG   HH
*              R7   R6   R5   R4   R3   R2   R1   M
*
*   OR
*
*              ERR DBXX AT YYYY  LINE ZZ  *
*
*
*              AA   BB   CC   DD   EE   FF   GG   HH
*              R7   R6   R5   R4   R3   R2   R1   M
*
*     WHERE
*
*         MB   = MOTHER BOARD
*         DB   = DAUGHTER BOARD
*         XX   = TEST LABLE. SEE JUMP TABLE AT LOCATION "HDTSA"
*                FOR A LIST OF TESTS PERFORMED.
*                EACH TEST NAME IS SUFFIXED BY -XX.
*         YYYY = ERROR LOCATION IN LISTING. HAS COMMENT
*                GIVING FAILING FUNTION.
*         ZZ   = LINE NUMBER
*
*  FOR SPECIALIST USAGE:
*          R1-R7, M ARE  CONTENTS OF REGISTERS.
*
*          AA - HH ARE CHANNEL PROGRAM OPERATION RESULTS.
*
*         IN ALL CASES:
*
*              R3 = CHANNEL NUMBER
*
*
*         IN GENERAL
*
*              R6 = SHOULD BE DATA
*              R5 = ACTUAL DATA
*              R7 = WORD NUMBER IN BLOCK TRANSFER
*
*
*         MLCP CHANNEL PROGRAM INFORMATION IS AS FOLLOWS
*
*              AA = CCP P VALUE, RCV
*              BB = CCP P VALUE, XMIT
*              CC = LR5 RCV/ LR5 XMIT
*              DD = ERROR CODE/LAST LR2 OUTPUT
*              EE = RCV LR7/CCP FLAG
*              FF = RCV CHAR COUNT (NEG)/XMIT CHAR COUNT (NEG)
*              GG = RCV FRAME (NEG)/XMIT FRAME (NEG)
*              HH = RCV CONFIG/ SIZE LAST XMIT BYTE
*
*              IF THE LEFT 8 BITS OF "DD" ARE NON ZERO THE MLCP
*              MLCP CHANNEL PROGRAM DETECTED AN ERROR. THE VALUE
*              SHOULD BE INTERPRETED FROM THE FOLLOWING TABLE:
*
*
*   1     LR7 STATUS WRONG AFTER INITIAL ILS RUPT
*   2     LR5 ADAPTER READY BIT NOT SET FOR RCV AFTER INITIAL ILS RUPT
*   3     BART DOESN'T BRANCH AFTER INITIAL ILS RUPT
*   4     LR5 ADAPTER RDY BIT SET WHEN SHOULD BE 0.
*   5     MISSED CLOSE FLAG, RANGE COMPLETED
*   6     NO ILS RUPT AFTER LAST FRAME.
*   7     MISSING ILS RUPT BETWEEN FRAMES
*   8     ADAPTER READY NOT SET AFTER FIRST CRI OF 2ND, 3D, OR 4TH FRAME.
*   9     ADAPTER READY SET WHEN SHOULD BE CLEARED
*   A     LR7 ERROR IN LB 0,1,2 FOR RCV AT EOF.
*   B     PENDING RECEIVE REQUEST AFTER LAST MESSAGE
*   C     PENDING TRANSMIT REQUEST AFTER TRANSMIT SHUT OFF
*   D    TWO SEQUENTIAL INPUT LR7  INSTRUCTIONS GAVE DIFFERENT RESULTS.
*   E     DELAY XMIT BIT SET DIDN'T INHIBIT DATA TRANSMISSION.
*   10    READ CRC STATUS SET WHEN NOT IN DIAGNOSTIC MODE.
*
*
*
*   A PROGRAM HALT WITH P DISPLAYING  A VALUE LESS THAN
*   HEX 100 INDICATES THAT A TRAP OCCURED. P DISPLAYS ONE
*   BEYOND THE ACTUAL TRAP VECTOR WHICH WAS TAKEN. THE
*   TRAP SAVE AREA IS FROM LOC. 2 - 9. REFER TO MODEL
*   34/36 T + V OPERATING INSTRUCTION MANUAL, DOC
*   71010460-200, SECTION 1.1 .
*
*
*   FOR SPECIALIST USAGE THERE IS A TABLE OF CHANNEL PROGRAM (CCP)
*   PARAMETERS WHICH IS FILLED AT THE TIME OF AN ERROR. (OUTTAB)
*
*
*
****************************************
****************************************
*
* RESTRICTIONS;
*
*    THIS PROGRAM REQUIRES A SYSTEM CONSOLE IF IT IS RUN ON A
* SYSTEM WHICH DOES NOT HAVE THE FULL CONTROL PANEL.
*
*    BHCLA FIRMWARE IS AT REV 1 AT THE INITIAL RELEASE OF
* THIS PROGRAM.
*
***************************************
*
*
*
* SOURCE = DCMS3A3
*
*        STANDARD REGISTER ASSIGNMENTS
*
*              $B5            V$LIB ENTRY
*              $B2            ENTRY TO MAJOR TEST
*              $B4,$B1        ENTRY TO SUB-ROUTINES
*E     $B7E        ENTRY TO ERROR ROUTINE
*
*E     $R3            CHANNEL
```

```
*E      $R5E        'IS' DATA WORD
*E      $R6E        'SHOULD BE' DATA WORD
*E      $R7E        TABLE COUNT
*
*
*****************************************************************
*
* STATUS BITS SET BY CCP IN MLCP STATUS
*
*
* BIT 02 DATA SERVICE ERROR  (X'2000')
* BIT 06 ABORT RECEIVED  (X'0200')
* BIT 09 CRC ERROR  (X'0040)
* BIT 11 EOF RECEIVED (X'0010')
*
*
*    FOR RECEIVE THESE BITS ARE
*   MAPPED FROM BITS 2,1,3, AND 0 OF LR7
*
*******************************************************************
*
*     TEST STRUCTURE (FOR SPECIALIST USE ONLY)
*
*
* CHANNEL PROGRAM INFORMATION
*
* DEFINITION OF CONTROL FLAG PASSED TO CCP
*
*    BIT 7      INSERT 100 MS IN XMIT CCP AFTER CHAR "N" IS TRANSMITTED.
*    BIT 6      SEND "ABORT" AFTER CHAR "N" IS TRANSMITTED.
*    BIT 5      TURN ON RCV AFTER FRAME "X" IS TRANSMITTED
*    BIT 4      READ IN CRC INFORMATION
*    BIT 3      INSERT 100  MS IN RCV CCP AFTER CHAR "N" IS INPUT.
*    BIT 2      INSERT DELAY BETWEEN FRAMES
*    BIT 1      ISSUE RCV RESYNC (OUT LR3)
*    BIT 0      ACCUMULATE CHAR. COUNT. IF = 0, CCP'S USE MINIMUM DATA LOOP
*
*
*    "X" AND "N" ARE FRAME AND CHAR NUMBERS PASSED TO THE CCPS BEFORE
*    EXECUTION TO SPECIFY AT WHAT TIME THE ACTION SPECIFIED BY THE ABOVE
*    FLAG IS TO BE TAKEN. THE FIRST CHARACTER
*    AND FRAME ARE NUMBERED "1".
*
*
*     THE PERTINANT LCT PARAMETERS ARE:
*
*    LOC        DESCRIPTION
*
*    3          SIZE OF LAST BYTE
*    4          SIZE OF RESIDUE EXPECTED
*    E          LR5 STORAGE (RCV)
*    14         LR2 STORAGE (RCV + XMIT)
*    17         LR6, RCV
*    18         CHAR COUNT, RCV, (-).
*    19         CONTROL FLAG FRAME NO., XMIT
*    1A         LR7, RCV
*    1B         FRAME COUNT, RCV (-)
*    1C         RCV TEMPORARY STORAGE
*    1D         CONTROL FLAG CHAR NO., RCV
*    1E         XMIT FW REV
*    1F         RCV FW REV
*    2E         LR5 STORAGE, XMIT
*    37         LR7, XMIT
*    38         CHAR COUNT, XMIT, (-)
*    39         TEMPORARY STORAGE, XMIT
*    3A         ERROR CODE
*    3B         FRAME COUNT, XMIT, (-)
*    3C         CONTROL FLAG CHAR NO., XMIT
*    3D         WORD SIZE MASK
*    3E         CONTROL FLAG (SEE ABOVE DESCRIPTION)
*    3F         BIT 0 = INTER FRAME FILL STATE (0 = ABORTS)
*               BIT 1 = DELAYED TRANSMIT  OF FIFO
*               BIT 2 = INITIAL FILL STATE (0 = ABORTS)
*               BITS 3 - 7 = CONTROL FLAG FRAME COUNT, RCV
*
*
*
*
*******************************************************************
*
*     CALLING SEQUENCE
*
* A COMMON CALLING SEQUENCE IS USED FOR DIFFERENT TESTS THROUGHOUT
* THE PROGRAM. IT IS:
*
*    LNJ   $B2,XXX    CALL TO SPECIFIC TEST
*    DC    A          RCV CONFIG (LR6)
*    DC    B          XMIT CONFIG (LR7)
*    DC    C          BIT 0 = INTER FRAME FILL MODE (0= ABORT)
*                     BIT 1 = DELAY XMIT BIT
*                     BIT 2 = INITIAL FILL MODE (0 = ABORT)
*                     BITS 5 - 7 = BYTE SIZE OF LAST BYTE
*                       (1 - 7 = 1 - 7 BITS, 0 = 8 BITS)
*    DC    D          INITIAL LR2 VALUE
*    DC    E          CCP CONTROL FLAG. (SEE PRIOR DESCRIPTION)
*    DC    F          BITS 0-7 = RCV FRAME FOR WHICH CONTROL FLAG IS VALID
*                     THIS IS PASSED AS A NEGATIVE QUANTITY.
*                     BITS 8 - 15 = CHAR NUMBER IN FRAME FOR WHICH
*                     CONTROL FLAG IS VALID.
*                     THIS IS ALSO PASSED AS A NEGATIVE QUANTITY.
*    DC    G          AS ABOVE BUT FOR XMIT
*
*    DC    H          1ST TRANSMIT BUFFER ADDRESS
*    DC    I          RANGE
*    DC    J          CONTROL WORD
*
*      .
*      .
*    DC    X          LAST TRANSMIT DATA BUFFER ADDRESS
*    DC    Y          RANGE
*    DC    Z          CONTROL WORD ( CONTAINS LAST BLOCK BIT)
*
*
```

```
                                   *   UP TO 4 DATA BLOCKS CAN BE SPECIFIED
                                   *
                                   **************
                                   *
                                   *   SPECIAL DIAGNOSTIC FEATURE
                                   *
                                   *       A DIAGNOSTIC MODE EXISTS WHICH ALLOWS CRC AND CRC
                                   *       RESIDUES TO BE INPUT. THIS IS ACCOMPLISHED BY CONFIGURING
                                   *       A RECIEVE BYTE SIZE OF 1 (VIA LR 6) AFTER THE LAST DATA
                                   *       BYTE OF A FRAME HAS BEEN RECEIVED.
                                   *
                                   *
                                   *
                                   * TEST DESCRIPTION
                                   *
                                   * THERE ARE 4 BASIC TESTS TO WHICH PARAMETERS ARE PASSED
                                   * USING THE SEQUENCE DESCRIBED ABOVE. THEY ARE:
                                   *
                                   *       TONTST - TURN ON TEST. TESTS IDLE LINK STATE INTERRUPT,
                                   *           FLAG IDLE STATE, TRANSMIT WITH NO RECEIVE. PROGRAM
                                   *               READS AMD REPORTS BHDLC FIRMWARE REV.
                                   *
                                   *
                                   *       LPTS - LOOP DATA TEST. DATA IS TURNED AROUND AND CHECKED
                                   *           FOR SINGLE FRAME CASES.
                                   *
                                   *       RET- RECEIVE END TEST. MULTIPLE FRAMES (1-4) CAN BE SENT
                                   *           AND RECEIVED WITH DIFFERENT ENDING CONDITIONS.
                                   *           A FLAG IS PASSED TO THE TEST WHICH SPECIFIES THE
                                   *           STATE OF THE FRAMES TO BE RECEIVED. THIS FLAG, EXST, IS AS
                                   *           FOLLOWS
                                   *
                                   *               ABCD
                                   *
                                   *               A = 4  BITS SPECIFYING STATE OF FRAME 1
                                   *               B =    "  "                     "  "   2
                                   *               C =    "  "                     "  "   3
                                   *               D =    "  "                     "  "   4
                                   *
                                   *
                                   *               STATE VALUES ARE AS FOLLOWS:
                                   *
                                   *               1 = NORMAL ENDING CONDITION
                                   *               2 = OVERUN
                                   *               3 = NULL (BHCLA IGNORES FRAME), LAST FRAME.
                                   *                   PROGRAM CHECKS MLCP STATUS FOR "0".
                                   *               4 = NULL (BHCLA IGNORS FRAME, NO REPORT), NOT LAST FRAME
                                   *                   PROGRAM SKIPS STATUS AND GOES TO NEXT.
                                   *               5 = MISSING FRAME
                                   *               6 = ABORTED FRAME
                                   *               7 = ABORTED FRAME DURING WHICH OVERUN OCCURED
                                   *
                                   *
                                   *       SPTS - SPEED TEST - PROGRAM LOOPS A 2048 BYTE FRAME
                                   *           FRAME AND TIMES UNTIL END OF RANGE OR 4 SECONDS
                                   *           AND REPORTS TEST CLOCK SPEED.
                                   *
                                   *   THE ABOVE 4 TESTS ALL USE A COMMAN DRIVER (LOAD) TO
                                   *   OUTPUT LCT AND CHANNEL PROGRAM   INFO TO THE MLCP.
                                   *
                                   * THERE IS ONE ADDITIONAL BASIC TEST WHICH HAS NO PARAMETERS
                                   * PASSED TO IT.
                                   *
                                   *
                                   *       CBLP-E - CONNECTOR LOOP DATA SET STATUS TEST. THIS TEST
                                   *           LOOPS DATA SET CONTROL SIGNALS TO DATA SET STATUS
                                   *           SIGNALS TO CHECK OUTPUT DRIVERS ON THE BHDLC BOARD.
                                   *           AN EXTERNAL WRAP CONNECTOR IS NECESSARY FOR THIS TEST.
                                   *
                                   *
                                   *
                                   *   ADDRESS, CONTROL, + LCF ARE FORMATTED FOR THE FIRST FRAME OF
                                   *   EACH TEST BY SUBROUTINE "GHEAD". SUBSEQUENT FRAMES IN EACH TEST
                                   *   HAVE ADDRESS AND CONTROL FIELDS SET TO 0.
                                   *
                                   *
                                   ******************************************************************* ***
                                   *
            0000                   ZERO     EQU     $
                                            XLOC    ZHNISA,ZHTH15,ZV$HR
                                            XLOC    ZV$RD,ZV$FI,ZV$FR,ZV$TTY
                                            XLOC    ZHISAZ,ZHIAFB,ZV$T
                                            XLOC    ZHWDTC,ZHRTCC
                                            XLOC    ZHRTCI,ZHRTCL
                                            XLOC    ZHPFR,ZHCOMM
                                            XLOC    ZV$PCH
                                            XLOC    ZV$BRK,ZV$BKF
                                            XLOC    ZV$QC,ZV$TC,ZV$TD,ZV$THZ
                                            CTRL    LINK ZV$TH
                                   *
                                   *
                                   *
                                   *
                                   *
                                   *
     00FF                                   ORG     ZERO+X'FF'
     00FF   0F06                   STOP     NOP     >RESTRT                 NO ERROR FOUND IN MLCC
                                   *
                                   *
     0100   89C0 1063              STRT     CMZ     FRST                    CHECK FIRST  TIME FLAG
     0102   0981 0150                       BNE     STLOOP                  BRANCH IF NOT FIRST TIME
     0104   0F7C                            NOP     >STRT
                                   *
     0105   8700 1182              RESTRT   CL      <PASSC                  CLEAR PASS COUNT
     0107   8700 1175                       CL      <IFLG                   CLEAR BIT INVERSION FLAG
                                   *
                                            CALL    ZV$RD,TITLE
     0109   FBC0 0003
     010B   D380 0000      X
     010D   0F80
     010E   11C1
     010F   9800 025D                       LDR     $R1,<NOP
     0111   9F00 00FF                       STR     $R1,<STOP
```

```
                                   *
                                   * ASK FOR WDT FREQUENCY (IF 6/3X)
                                   *
0113  8980 1164                            CMZ    <FRST                   CHECK FIRST TIME FLAG
0115  0980 013E                            BNE    <RST
0117  8A80 1164                            INC    <FRST                   SET FIRST TIME FLAG
0119  1C3C                                 LDV    $R1,=60                 DEFAULT FREQUENCY
011A  9F00 1191                            STR    $R1,<HRTZ
011C  8C51                                 STS    =$R1
011D  82D1                                 LB     =$R1,=Z'2000'
011E  2000
011F  0501 000D                            BBT    RFRIQ                   B = 6/4X
                                   FREQ    CALL   ZV$QC,MESG5             PWR FREQ (HZ)
0121  FBC0 0003
0123  D380 0000        X
0125  0F80
0126  11E1
                                           CALL   ZV$ID,HRTZ              INPUT
0127  FBC0 0003
0129  D380 0000        X
012B  0F80
012C  1191
012D  8756                         RFRIQ   CL     =$R6
                                   * CONVERT WDT TICK TO U SEC
012E  F870 C350                            LDR    $R7,=Z'C350'            50000
0130  F300 1191                            DIV    $R7,<HRTZ
0132  8756                                 CL     =$R6
0133  F370 0064                            DIV    $R7,=100
0135  FF00 1193                            STR    $R7,<DIV1
0137  70D0                                 DOR    $R7,16
0138  F370 000A                            DIV    $R7,=10
013A  FF00 1194                            STR    $R7,<DIV2
013C  EF00 1195                            STR    $R6,<DIV3
013E  4C02                         RST     LDV    $R4,=2
013F  CF00 116C                            STR    $R4,<RANGE
0141  1CFF                                 LDV    $R1,=-1
0142  9F00 11B9                            STR    $R1,<ATLT
0144  9F00 11BA                            STR    $R1,<ATLT+1             INITIALIZE ACTIVE LINE TABLE
0146  9F00 11BB                            STR    $R1,<ATLT+2
                                           CALL   ZV$F,TMPSTR,C0,C8
0148  FBC0 0003
014A  D380 0000        X
014C  0F80
014D  11AE
014E  118E
014F  118F
0150  9800 1171                            LDR    $R1,<DADD               GET LAST DEVICE ADDRESS GIVEN
0152  9F00 11AE                            STR    $R1,<TMPSTR             USE AS DEFAULT
                                   *
                                   * ASK FOR BBHDLC ADDRESS
                                   *
                                           CALL   ZV$QC,MESG1
0154  FBC0 0003
0156  D380 0000        X
0158  0F80
0159  11D5
                                           CALL   ZV$IH,TMPSTR,RANGE
015A  FBC0 0003
015C  D380 0000        X
015E  0F80
015F  11AE
0160  116C
0161  8752                                 CL     =$R2
0162  9800 11AE                            LDR    $R1,<TMPSTR
0164  1901 0020                            BEZ    $R1,JP2                 IF ZERO, INVALID
0166  9F00 1171                            STR    $R1,<DADD               NEW DEFAULT
0168  AB80 11AE                            LAB    $B2,<TMPSTR
016A  CB80 019E                            LAB    $B4,<NODEVF
016C  CF80 0000        X                   STB    $B4,<ZHTH15
016E  986E                         SALT    LDR    $R1,$B2.+$R2            GET VALUE INPUT
016F  1900 01DD                            BEZ    $R1,<CON3               CHECK FOR END OF LIST
0171  9970 FFC0                            CMR    $R1,=Z'FFC0'            COMPARE CHANNEL WITH HEX FFC0
0173  030A                                 BG     >JP1                    CHANNEL NUMBER TO LARGE
0174  9970 03C0                    RCK     CMR    $R1,=X'3C0'             COMPARE CHANNEL WITH HEX 3C0
0176  0387                                 BLE    >JP1                    CHANNEL NUMBER TO SMALL
0177  1E26                         CON1    ADV    $R1,=X'26'              PUT FUNCTION IN
0178  8055                                 IO     =$R5,=$R1               INPUT DEVICE NUMBER
0179  0051
017A  D970 2178                            CMR    $R5,=Z'2178'            COMPARE DEVICE NUMBER RESPONSE
017C  0924                                 BE     >CON2
017D  1D01                         JP1     CMV    $R1,=1
017E  0987                                 BNE    >JP2
                                           CALL   ZV$PCH                  GO TO PATCHER IF ADD = 1.
017F  FBF0 0001
0181  D380 0000        X
0183  0F80 013E                            B      <RST                    RETURN
0185  9870 2A2A                    JP2     LDR    $R1,=A'**'
0187  9F00 11EA                            STR    $R1,<ERMG+1
                                           CALL   ZV$TC,BADINP
0189  FBC0 0003
018B  D380 0000        X
018D  0F80
018E  0191
018F  0F80 013E                            B      <RST
0191  4248 434C 4120               BADINP  TEXT   'BHCLA NOT ON THIS CHANNEL$'
0194  4E4F 5420 4F4E
      2054 4849 5320
      4348 414E 4E45
      4C24
                                   *
019E  8755                         NODEVF  CL     =$R5                    TRAP HANDLER ROUTINE
019F  0003                                 RTT
                                   *
                                   *
                                   *
                                   *         PREPARE ACTIVE LINE TABLE
                                   *
                                   *
01A0  1E02                         CON2    ADV    $R1,=2                  FORM FC FOR EXTENDED ID
01A1  8055                                 IO     =$R5,=$R1               READ EXTENDED ID
01A2  0051
01A3  0703                                 BIOT   >$+2+$AF
01A4  E380 0FBA                            LNJ    $B6,<ERRMB              READ EXTENDED ID WAS NAK'ED
```

```
01A6  5048                     SOR    $R5,8              ALIGN EXTENDED ID
01A7  D470 2100                OR     $R5,=Z'2100'
01A9  DF00 116D                STR    $R5,<DEVID
                        *
01AB  D970 21F6                CMR    $R5,=Z'21F6'
01AD  09A9                     BNE    >JP3               BR = NOT BH4DLD
01AE  9570 03C0         CON4   AND    $R1,=X'3C0'
01B0  1047                     SOR    $R1,7              GET LINE NUMBER
01B1  1D04                     CMV    $R1,=4             CHECK FOR LEGAL LINE VALUE
01B2  0353                     BG     >JP2               BRANCH IF NO GOOD
                        *
01B3  9F00 1177                STR    $R1,<TPR
01B5  1041                     SOR    $R1,1              GET LINE ADAPTER NUMBER
01B6  DF10 11B9                STR    $R5,<ATLT,$R1      STORE ID
                        *
01B8  9800 1177                LDR    $R1,<TPR
01BA  9F00 1C00                STR    $R1,<RTB
                               CALL   ZV$TC,IDMSG        LINE
01BC  FBC0 0003
01BE  D380 0000      X
01C0  0F80
01C1  11F7
                               CALL   ZV$TD,RTB          NUMBER
01C2  FBC0 0003
01C4  D380 0000      X
01C6  0F80
01C7  1C00
                               CALL   ZV$T,EQ            =
01C8  FBC0 0003
01CA  D380 0000      X
01CC  0F80
01CD  11FB
                               CALL   ZV$TH,DEVID        DEVICE ID
01CE  FBC0 0003
01D0  D380 0000      X
01D2  0F80
01D3  116D
                        *
01D4  0F80 016E                B      <SALT
                        *
01D6  D970 21F7         JP3    CMR    $R5,=Z'21F7'
01D8  0980 0185                BNE    <JP2               B = NOT BH4DLE
01DA  7C00                     LDV    $R7,=0
01DB  0F80 01AE                B      <CON4
                        *
                        * MODIFY LIST OF IO CONTROL WORDS
                        *
                        *
                        *
01DD  2CF4              CON3   LDV    $R2,=-12           NUMBER OF I/O TO BE CHANGED
                               CALL   ZV$IZ              INITIALIZE TRAPS AGAIN
01DE  FBF0 0001
01E0  D380 0000      X
01E2  9800 11AE                LDR    $R1,<TMPSTR        GET FIRST ADDRESS INPUT !
01E4  DB80 11AE                LAB    $B5,<CONT1+12      PUT ADDRESS OF CONTROL TABLE IN $B5
01E6  C870 003F         ALL    LDR    $R4,=X'3F'         LOAD MASK TO CLEAR CHANNEL
01E8  9570 FC00                AND    $R1,=Z'FC00'       CLEAR SUBCH. & FUNCTION
01EA  C525                     AND    $R4,$B5,$R2        CLEAR CHANNEL
01EB  9454                     OR     $R1,=$R4           PUT CHANNEL NUMBER IN
01EC  9F25                     STR    $R1,$B5,$R2        STORE IT BACK IN CONTROL TABLE
01ED  27F9                     BINC   $R2,>ALL
                        *
                        * ASK FOR LOOP MODE - C = CONNECTOR LOOP, A = LINE ADAPT
                        *
01EE  8700 117E         QSTR   CL     <QFLG              CLEAR "QUICK FLAG"
01F0  8700 117F                CL     <PFLAG             CLEAR PRINT FLAG
01F2  8700 1184                CL     <XLOOP             CLEAR EXT LOOP FLAG
01F4  8700 1185                CL     <CBLOOP            CLEAR CONNECTOR LOOP FLAG
                        RUTBG  CALL   ZV$QC,MESG2        ASK FOR MODE
01F6  FBC0 0003
01F8  D380 0000      X
01FA  0F80
01FB  11DE
                               CALL   ZV$IA,MASK,TPR
01FC  FBC0 0003
01FE  D380 0000      X
0200  0F80
0201  1176
0202  1177
0203  9800 1177                LDR    $R1,<TPR           GET INPUT
0205  1048                     SOR    $R1,8
0206  9970 0043                CMR    $R1,=X'43'         C
0208  0F00 01F6                NOP    <RUTBG
020A  0980 0210                BNE    <RUT1
020C  8A80 1185                INC    <CBLOOP            SET CONNECTOR LOOP FLAG
020E  0F80 023E                B      <RUT4
                        *
0210  9970 0041         RUT1   CMR    $R1,=X'41'         A
0212  0900 023E                BE     <RUT4
0214  9970 0045                CMR    $R1,=X'45'         E
0216  0980 021C                BNE    <RUTEND
0218  8A80 1184                INC    <XLOOP             SET EXTERNAL LOOP FLAG
021A  0F80 023E                B      <RUT4
                        *
021C  0F00 0210         RUTEND NOP    <RUT1
021E  9970 0051                CMR    $R1,=X'51'         (Q)UICK PASS
0220  0980 0226                BNE    <RUTBGA
0222  8A80 117E                INC    <QFLG
0224  0F80 01F6                B      <RUTBG
0226  9970 0050         RUTBGA CMR    $R1,=X'50'         P
0228  0980 022E                BNE    <RUTBGB
022A  8A80 117F                INC    <PFLAG             SET TO PRINT TEST LABLES
022C  0F81 FFC9                B      RUTBG
022E  9970 0049         RUTBGB CMR    $R1,=X'49'         I
0230  0900 0105                BE     <RESTRT
                        *
                               CALL   ZV$TC,NOGO         INVALID
0232  FBC0 0003
0234  D380 0000      X
0236  0F80
0237  023A
0238  0F81 FFBD                B      RUTBG
                        *
```

```
023A    494E 5641 4C49                 NOGO    TEXT     'INVALID$'
023D    4424
                                       *
023E    9870 2A2A                      RUT4    LDR      $R1,=A'**'
0240    9F00 11EA                              STR      $R1,<ERMG+1
                                               CALL     ZV$TC,MESG3             PRINT MLCP
0242    FBC0 0003
0244    D380 0000               X
0246    0F80
0247    0250
0248    8753                                   CL       =$R3                    SET CHANNEL NUMBER TO 0
0249    C380 0ED0                              LNJ      $B4,<FLN                FIND ACTIVE LINE NUMBER
024B    0000                                   HLT                              IMPOSSIBLE, INDICATES NO BBHDLC
024C    E380 0C9A                              LNJ      $B6,<PREV               PRINT FIRMWARE REV NUMBER
024E    0F80 0253                              B        <STLOOP
0250    4D4C 4350 2020                 MESG3   TEXT     'MLCP  '
                                       *
                                       *
                                       * JUMP TABLE FOR TESTS
                                       *
                                       *
0253    9800 025D                      STLOOP  LDR      $R1,<NOP
0255    9F00 00FF                              STR      $R1,<STOP
0257    8700 1179                              CL       <LOOP                   SET LOOP COUNT TO 0
0259    8700 1199                              CL       <SEC                    SECONDS
025B    8700 1198                              CL       <TTOT
                                       *
025D    0F76                           NOP     NOP      >STLOOP
025E    E380 02DE                      HDTSA   LNJ      $B6,<TON-AA             TURN ON TESTS
0260    0F7E                                   NOP      >HDTSA
0261    E380 02F1                              LNJ      $B6,<LPTS-A             SIMPLE DATA LOOP
0263    0F7B                                   NOP      >HDTSA
0264    E380 07EE                              LNJ      $B6,<SPED-W             DO SPEED TEST.
0266    0F78                                   NOP      >HDTSA
0267    E380 0384                              LNJ      $B6,<CBLP-E             CONNECTOR LOOP STATUS TEST
0269    0F75                                   NOP      >HDTSA
026A    9800 1192                              LDR      $R1,<SPEED              GET SPEED/10
026C    9970 1770                              CMR      $R1,=6000
026E    0201 0006                              BL       LOSPD                   B = LOW SPEED
0270    1C01                                   LDV      $R1,=1
0271    9F00 1161                              STR      $R1,<SPFLG              SET FLAG FOR 72 KB
0273    0F80 02B7                              B        <HSPD
                                       *
0275    8700 1161                      LOSPD   CL       <SPFLG
0277    E380 0312                              LNJ      $B6,<LPTS-B             5 BIT CHAR
0279    0F65                                   NOP      >HDTSA
027A    E380 032D                              LNJ      $B6,<LPT-BA             70 CHAR, 5 BIT MODE
027C    0F04                                   NOP      >$+4
027D    E380 034E                              LNJ      $B6,<LPTS-C             6 BIT CHAR.
027F    0F5F                                   NOP      >HDTSA
0280    E380 0369                              LNJ      $B6,<LPTS-D             7 BIT CHAR
0282    0F5C                                   NOP      >HDTSA
0283    E380 0408                              LNJ      $B6,<TRAN-F             TRANSPARENCY TEST 1
0285    0F59                                   NOP      >HDTSA
0286    E380 042E                              LNJ      $B6,<SUP-G              TEST SUPERVISORY CONTROL BIT
0288    0F56                                   NOP      >HDTSA
0289    E380 044B                              LNJ      $B6,<PRT-GA             TCB, CFX, AFX TESTS
                                       *                                        INCLUDES TESTS GA TO GR.
028B    0F53                                   NOP      >HDTSA
                                       *
028C    E380 04B6                              LNJ      $B6,<MULT-H             2 FRAMES BACK TO BACK
028E    0F50                                   NOP      >HDTSA
028F    E380 04D8                              LNJ      $B6,<MULT-I             2 FRAMES, FLAGS BETWEEN
0291    0F04                                   NOP      >$+4
0292    E380 04FB                              LNJ      $B6,<MUL-IA             2 FRAMES, DELAYED TRANSMISSION MODE
0294    0F04                                   NOP      >$+4
0295    E380 051E                              LNJ      $B6,<TERM-J             SINGLE FRAME  TERMINATION TESTS
                                       *                                        INCLUDES  TESTS JA TO JK
0297    0F01 0001                              NOP      $+2
0299    E380 0593                              LNJ      $B6,<MULT-L             2 FRAMES, ABORTS BETWEEN
029B    0F04                                   NOP      >$+4
029C    E380 05B6                              LNJ      $B6,<UND1-K             UNDERUN, ABORT TEST 1
029E    0F04                                   NOP      >$+4
029F    E380 05D1                              LNJ      $B6,<RCV-EA             RECEIVE END TEST 1
02A1    0F04                                   NOP      >$+4
02A2    E380 05F0                              LNJ      $B6,<RCV-EB             RECEIVE END TEST 2
02A4    0F04                                   NOP      >$+4
02A5    E380 060F                              LNJ      $B6,<RCV-EC             RECEIVE END TEST 3
02A7    0F2A                                   NOP      >PSPT
02A8    E380 062E                              LNJ      $B6,<RCV-ED             RECEIVE END TEST 4
02AA    0F27                                   NOP      >PSPT
02AB    E380 064D                              LNJ      $B6,<TFR-M              DO TWO FRAME TESTS
                                       *                                        INCLUDES TESTS MA - M7
02AD    0F04                                   NOP      >$+4
02AE    E380 077F                              LNJ      $B6,<RCV-EN             RECEIVE END TEST
02B0    0F21                                   NOP      >PSPT
02B1    E380 07A4                              LNJ      $B6,<RCV-EO             RECEIVE END TEST
02B3    0F1E                                   NOP      >PSPT
02B4    E380 07C9                              LNJ      $B6,<RCV-EP             RECEPVE END TEST
02B6    0F1B                                   NOP      >PSPT
02B7    E380 0817                      HSPD    LNJ      $B6,<RAN-RD             RANDOM DATA TEST
02B9    0F04                                   NOP      >$+4
02BA    8980 1161                              CMZ      <SPFLG
02BC    0300 02C0                              BG       <BYPSS                  B = 72 KB
                                       *
02BE    E380 0856                              LNJ      $B6,<PCRC-X             PARTIAL BYTE, CRC TEST
02C0    0F04                           BYPSS   NOP      >$+4
02C1    8A80 1179                              INC      <LOOP                   BUMP LOOPCOUNT
02C3    8980 117E                              CMZ      <QFLG                   CHEC QUICK FLAG
02C5    0980 01EE                              BNE      <QSTR                   B = QUICK PASS
02C7    0F0A                                   NOP      >PSPT
02C8    8980 0000               X              CMZ      <ZV$TTY                 CHECK IF TTY ON SYSTEM
02CA    0981 0006                              BNE      PSPT                    B = IS CONSOLE
02CC    1C00                                   LDV      $R1,=0
02CD    9F00 00FF                              STR      $R1,<STOP               INSERT EOP HALT
02CF    0F80 00FF                              B        <STOP
                                       PSPT    CALL     ZV$TC,PASMSG
02D1    FBC0 0003
02D3    D380 0000               X
02D5    0F80
02D6    02DB
02D7    8A80 1182                              INC      <PASSC                  BUMP PASS COUNT
02D9    0F80 00FF                              B        <STOP
```

```
02DB  5041 5353 2024          PASMSG TEXT   'PASS $'
                              *----------------------------------------------------------------
                              *
                              * INITIAL TURN ON TESTS
                              *
02DE  9870 4141               TON-AA LDR    $R1,=A'AA'
02E0  9F00 11EA                      STR    $R1,<ERMG+1          REPORT TEST SECTION
                              *
02E2  8700 116A                      CL     <CHSZ                SET FOR 8 BIT BYTES
02E4  A380 0AF2                      LNJ    $B2,<TONTST          INITIAL TURN ON TEST
02E6  000C                           DC     X'C'                 RCV CONFIG, 8 BITS, LR 6
02E7  000C                           DC     X'C'                 XMIT CONFIG, 8 BITS, LR7
02E8  000C                           DC     X'C'                 FILL MODE, ABORT IDLE
02E9  0045                           DC     X'45'                LR2, TEST, XMIT ON
02EA  0080                           DC     X'80'                ACTION FLAG
02EB  0000                           DC     0                    RCV ACTION  FRAME, CHAR
02EC  0000                           DC     0                    XMIT ACT FRAME, CHAR
02ED  1800                           DC     <SDB                 DATA ADDRESS
02EE  0005                           DC     5                    RANGE
02EF  0060                           DC     X'60'                LAST BLOCK, VALID
                              *
02F0  8386                           JMP    $B6                  EXIT
                              *
                              *****************************************************************
                              *
                              *  - SIMPLE DATA TEST
                              *
02F1  9870 2041               LPTS-A LDR    $R1,=A' A'
02F3  9F00 11EA                      STR    $R1,<ERMG+1          REPORT TEST
                                     CALL   ZV$F,SDB,PAT,C8      FILL 8 WORDS OF 0303
02F5  FBC0 0003
02F7  D380 0000         X
02F9  0F80
02FA  1800
02FB  117D
02FC  118F
02FD  8700 1174                      CL     <HEAD                CLEAR HEADER FLAG
02FF  AB80 0307                      LAB    $B2,<PAR1
0301  AF80 0ECF                      STB    $B2,<PARPTR          STORE PARAMETER POINTER
0303  C380 0D80                      LNJ    $B4,<GHEAD           GENERATE HEADER
                              *
0305  A380 08CD                      LNJ    $B2,<LPTS            LOOP TEST
0307  000C                    PAR1   DC     =X'C'                LR6, RCV CFG, 8 BITS
0308  000C                           DC     X'C'                 LR7, XMIT , 8 BITS
0309  000C                           DC     X'C'                 FILL STATE, ABORT IDLE
030A  0047                           DC     =X'47'               LR2 CONTROL, RCV, XMIT, TEST
030B  0000                           DC     X'0'                 ACTION FLAG
030C  0000                           DC     0                    RCV ACTION  FRAME, CHAR
030D  0000                           DC     0                    XMIT ACT FRAME, CHAR
030E  1800                           DC     <SDB                 DATA
030F  0010                           DC     16                   RANGE
0310  0060                           DC     X'60'                LAST BLOCK, VALID
                              *
0311  8386                           JMP    $B6
                              *----------------------------------------------------------------
                              *
                              * LOOP TEST B - 5 BIT CHARACTERS, ASCENDING DATA
                              *
0312  9870 2042               LPTS-B LDR    $R1,=A' B'
0314  9F00 11EA                      STR    $R1,<ERMG+1          REPORT TEST
0316  B380 0D3D                      LNJ    $B3,<FACDTA          FILL ASCENDING DATA
                              *
0318  AB80 0322                      LAB    $B2,<PAR2
031A  AF80 0ECF                      STB    $B2,<PARPTR
031C  8700 1174                      CL     <HEAD                CLEAR HEADER FLAG
031E  C380 0D80                      LNJ    $B4,<GHEAD           GENERATE HEADER
                              *
0320  A380 08CD                      LNJ    $B2,<LPTS
0322  0000                    PAR2   DC     0                    5 BITS, LR6, RCV
0323  0000                           DC     X'0'                 5 BITS, XMIT, LR7
0324  0000                           DC     0                    FILL MODE = ABORT
0325  0047                           DC     =X'47'               LR2 - RCV, XMIT, TEST
0326  0000                           DC     X'0'                 ACTION FLAG
0327  0000                           DC     0                    RCV ACTION  FRAME, CHAR
0328  0000                           DC     0                    XMIT ACT FRAME, CHAR
0329  1800                           DC     <SDB                 DATA
032A  0020                           DC     X'20'                RANGE
032B  0060                           DC     X'60'                LAST,V
                              *
032C  8386                           JMP    $B6
                              *
                              *----------------------------------------------------------------
                              *
                              * TEST USING 5 BIT CHAR AND RANGE > FIFO LENGTH
                              *
032D  9870 4241               LPT-BA LDR    $R1,=A'BA'
032F  9F00 11EA                      STR    $R1,<ERMG+1          REPORT TEST
                                     CALL   ZV$F,SDB,PAT,C70     FILL 70 WORDS OF 0303
0331  FBC0 0003
0333  D380 0000         X
0335  0F80
0336  1800
0337  117D
0338  1190
0339  AB80 0343                      LAB    $B2,<PAR3
033B  AF80 0ECF                      STB    $B2,<PARPTR
033D  8700 1174                      CL     <HEAD                CLEAR HEADER FLAG
033F  C380 0D80                      LNJ    $B4,<GHEAD           GENERATE HEADER
                              *
0341  A380 08CD                      LNJ    $B2,<LPTS            LOOP TEST
0343  0000                    PAR3   DC     =X'0'                LR6, RCV CFG, 5 BITS
0344  0000                           DC     0                    LR7, XMIT , 5 BITS
0345  0000                           DC     0                    FILL STATE, ABORT IDLE
0346  0047                           DC     =X'47'               LR2 CONTROL, RCV, XMIT, TEST
0347  0000                           DC     X'0'                 ACTION FLAG
0348  0000                           DC     0                    RCV ACTION  FRAME, CHAR
0349  0000                           DC     0                    XMIT ACT FRAME, CHAR
034A  1800                           DC     <SDB                 DATA
034B  0046                           DC     70                   RANGE
034C  0060                           DC     X'60'                LAST BLOCK, VALID
                              *
034D  8386                           JMP    $B6
                              *----------------------------------------------------------------
```

```
                     *
                     * LOOP TEST C - 6 BIT CHAR, ASCENDING DATA
                     *
034E  9870 2043      LPTS-C LDR    $R1,=A' C'
0350  9F00 11EA             STR    $R1,<ERMG+1
0352  B380 0D3D             LNJ    $B3,<FACDTA          FILL ASCENDING DATA
                     *
0354  AB80 035E             LAB    $B2,<PAR4
0356  AF80 0ECF             STB    $B2,<PARPTR
0358  8700 1174             CL     <HEAD                CLEAR HEADER FLAG
035A  C380 0D80             LNJ    $B4,<GHEAD           GENERATE HEADER
                     *
035C  A380 08CD             LNJ    $B2,<LPTS
035E  0004           PAR4   DC     4                    6, BITS, RCV LR 6
035F  0004                  DC     X'4'                 6 BITS, LR7, XMIT
0360  0004                  DC     4                    FILL MODE, ABORT
0361  0047                  DC     =X'47'               LR2, RCV, XMIT, TEST
0362  0000                  DC     X'0'                 ACTION FLAG
0363  0000                  DC     0                    RCV ACTION  FRAME, CHAR
0364  0000                  DC     0                    XMIT ACT FRAME, CHAR
0365  1800                  DC     <SDB                 DATA
0366  0040                  DC     =X'40'               RANGE
0367  0060                  DC     X'60'                LAST,V
                     *
0368  8386                  JMP    $B6
                     *---------------------------------------------------------------
                     *
                     * LOOP TEST D - 7 BIT CHARACTERS, ASCENDING DATA
                     *
0369  9870 2044      LPTS-D LDR    $R1,=A' D'
036B  9F00 11EA             STR    $R1,<ERMG+1          REPORT
036D  B380 0D3D             LNJ    $B3,<FACDTA          FILL ASCENDING DATA
                     *
036F  AB80 0379             LAB    $B2,<PAR5
0371  AF80 0ECF             STB    $B2,<PARPTR
0373  8700 1174             CL     <HEAD                CLEAR HEADER FLAG
0375  C380 0D80             LNJ    $B4,<GHEAD           GENERATE HEADER
                     *
0377  A380 08CD             LNJ    $B2,<LPTS            LOOP TEST
0379  0008           PAR5   DC     8                    7 BITS, LR6, RCV
037A  0008                  DC     X'8'                 7 BITS, LR7, XMIT
037B  0008                  DC     8                    ABORT IDLE, LAST = 7 BITS
037C  0047                  DC     X'47'                LR2, RCV, XMIT, TEST
037D  0000                  DC     X'0'                 ACTION FLAG
037E  0000                  DC     0                    RCV ACTION  FRAME, CHAR
037F  0000                  DC     0                    XMIT ACT FRAME, CHAR
0380  1800                  DC     <SDB                 DATA
0381  0080                  DC     X'80'
0382  0060                  DC     X'60'                LAST,V
                     *
0383  8386                  JMP    $B6
                     *---------------------------------------------------------------
                     *
                     * CONNECTOR LOOP STATUS TEST
                     *
                     *   SEE THE  PROGRAM HEADING FOR A DESCRIPTION OF LOOP-BACK
                     *   CONNECTORS FOR BH4DLD AND BH4DLE.
                     *
                     *
                     *   IN THE TEST A CHANNEL PROGRAM IS SET UP WHICH OUTPUTS THE
                     *   FOLLOWING 4 VALUES TO LR2.
                     *
                     *         X'00',X'80',X'40',X'3C'
                     *
                     *   THE FOLLOWING VALUES ARE INPUT FROM LR5.
                     *
                     *        BH4DLD     00,10,E0,0
                     *        BH4DLE     00,90,60,0
                     *
                     *
0384  9870 2045      CBLP-E LDR    $R1,=A' E'
0386  9F00 11EA             STR    $R1,<ERMG+1          REPORT TEST
0388  8F00 2063             SAVE   <SAVMAJ,=Z'0002'     SAVE B6
038A  0002
                     *
038B  9800 1185             LDR    $R1,<CBLOOP          GET CONNECTOR LOOP FLAG
038D  1980 0393             BNEZ   $R1,<CBLP1           BRANCH IF THERE IS A CONNECTOR LOOP
038F  8F80 2063      CBLP8  RSTR   <SAVMAJ,=Z'0002'
0391  0002
0392  8386                  JMP    $B6                  EXIT TEST, NO CONNECTOR LOOP
                     *
0393  C380 2000      CBLP1  LNJ    $B4,<PLB             PRINT TEST LABLE
0395  8753                  CL     =$R3                 SET FOR CHANNEL 0
0396  C380 0ED0      CBLP2  LNJ    $B4,<FLN             FIND ACTIVE LINE
0398  0FF7                  B      >CBLP8               NO MORE LINES
0399  C380 0EEC             LNJ    $B4,<GENITZ          DO GENERAL INITIALIZE
                     *
                     * SEND OUT CHANNEL PROGRAM
                     *
039B  3B80 039E             BODD   $R3,<CBLP7           BRANCH IF XMIT CHANNEL
039D  8AD3                  INC    =$R3                 MAKE XMIT CHAN
039E  9380 1095      CBLP7  LNJ    $B1,<SDATA
03A0  139A                  DC     <CCP4                CHANNEL PROGRAM
03A1  004E                  DC     (CCP5-CCP4)*2        RANGE
03A2  0200                  DC     X'200'               RAM ADDRESS
03A3  0000                  DC     0                    EVEN CPU ADDRESS
                     *
03A4  BF00 1177             STR    $R3,<TPR
                     *
                     * SEND OUT DATA FOR CHANNEL PROGRAM IF BH4DLE
                     *
03A6  3042                  SOR    $R3,2                GET LA NUMBER
03A7  9830 11B9             LDR    $R1,<ATLT,$R3        GET ID
03A9  9F00 116D             STR    $R1,<DEVID
03AB  B800 1177             LDR    $R3,<TPR
03AD  9970 21F7             CMR    $R1,=Z'21F7'
03AF  0905                  BE     >CHLPY               B = BH4DLE
                     *
03B0  B380 10EB             LNJ    $B3,<SETLCT
03B2  03F8                  DC     <LCT9                LCT FOR BH4DLD
03B3  0F84                  B      >CHLPX
03B4  B380 10EB      CHLPY  LNJ    $B3,<SETLCT
03B6  03FD                  DC     <LCT10               LCT FOR BH4DLE
```

```
                          *
                          *
                          * SEND OUT CHANNEL PROGRAM START
                          *
03B7  B380 10EB           CHLPX   LNJ    $B3,<SETLCT
03B9  0402                        DC     <LCT6
                          *
03BA  8751                CBLP9   CL     =$R1
03BB  C380 0F0B           CBLP5   LNJ    $B4,<CHCT            DO CHANNEL CONTROL
03BD  0F82                        B      >$+2
03BE  4000                        DC     Z'4000'              START IO
03BF  C380 1067                   LNJ    $B4,<DLAYLG          225 MS DELAY
                          *
                          * READ DATA SET STATUS
                          *
03C1  9B80 03F0                   LAB    $B1,<DS-SB
03C3  A800 116D                   LDR    $R2,<DEVID           GET ID
03C5  A970 21F6                   CMR    $R2,=Z'21F6'
03C7  0903                        BE     >CUR                 BRANCH = BH4DLD
03C8  9B80 03F4                   LAB    $B1,<DS-SB1
                          *
03CA  C380 0F9E           CUR     LNJ    $B4,<DSSTA
03CC  D570 F700                   AND    $R5,=Z'F700'         STRIP STATUS TO PERTINENT BITS
03CE  E811                        LDR    $R6,$B1.$R1          PICK UP SHOULD BE
03CF  D956                        CMR    $R5,=$R6
03D0  0900 03D4                   BE     <CBLP-3              BRANCH IF GOOD
03D2  E3C0 0BDB                   LNJ    $B6,ERRDB            DATA SET STATUS BAD
                          *
                          * READ LCT STAT - CHAN PROG SETS BIT 5 = 1 IF IT FAILED
                          *
03D4  C800 11AB           CBLP-3  LDR    $R4,<CONT10          INPUT LCT STATUS
03D6  BF00 1177                   STR    $R3,<TPR             STORE CHANNEL NUMBER
03D8  3B00 03DB                   BEVN   $R3,<CBLP6           BRANCH IF RECEIVE CHANNEL
03DA  88D3                        DEC    =$R3
03DB  C380 0EE8           CBLP6   LNJ    $B4,<CGSCH           FORM FC AND ADDRESS
03DD  8055                        IO     =$R5,=$R4            INPUT LCT STATUS TO R5
03DE  0054
03DF  0703                        BIOT   >$+2+$AF
03E0  E380 0FBA                   LNJ    $B6,<ERRMB           INSTRUCTION WAS NAK'ED
                          *
03E2  B800 1177                   LDR    $R3,<TPR             GET BACK TEST CHANNEL
03E4  8756                        CL     =$R6
03E5  5900 03E9                   BEZ    $R5,<CBLP4
03E7  E3C0 0BC6                   LNJ    $B6,ERRDB            ERROR, CHANNEL PROGRAM GOT
                          *                             WRONG DATA SET STATUS    .
03E9  8AD1                CBLP4   INC    =$R1
03EA  1D04                        CMV    $R1,=4
03EB  0201 FFCF                   BL     CBLP5                DO FOR NEXT VALUE
03ED  3E04                        ADV    $R3,=4               BUMP CHANNEL NUMBER
03EE  0F80 0396                   B      <CBLP2               TRY NEXT CHANNEL
                          * * TABLE OF VALUES TO BE RETURNED (BH4DLD)
                          *
03F0  0000                DS-SB   DC     Z'0000'
03F1  1000                        DC     Z'1000'
03F2  E000                        DC     Z'E000'
03F3  0000                        DC     Z'0'
                          *
                          * BH4DLE
                          *
03F4  0000                DS-SB1  DC     Z'0000'
03F5  9000                        DC     Z'9000'
03F6  6000                        DC     Z'6000'
03F7  0000                        DC     Z'0000'
                          *
                          *
03F8  103E                LCT9    DC     Z'103E'
03F9  E03F                        DC     Z'E03F'
03FA  0105                        DC     Z'0105'              PAUSE DISABLE, RECEIVE
03FB  0125                        DC     Z'0125'              DITTO, XMIT
03FC  0000                        DC     0
                          *
                          * LCT FOR BH4DLE
                          *
03FD  903E                LCT10   DC     Z'903E'
03FE  603F                        DC     Z'603F'
03FF  0105                        DC     Z'0105'              PAUSE DISABLE,RCV
0400  0125                        DC     Z'0125'              PAUSE DISABLE, XMIT
0401  0000                        DC     0
                          *
                          *
                          * LCT TABLE
                          *
0402  0206                LCT6    DC     Z'0206'              RCV P, MSB
0403  0007                        DC     Z'0007'              RCV P, LSB
0404  0226                        DC     Z'0226'              XMIT P, MSB
0405  0027                        DC     Z'0027'              XMIT P, LSB
0406  1037                        DC     Z'1037'              BYTE TO INPUT
0407  0000                        DC     0                    END OF LIST
                          *
                          *
                          *------------------------------------------------------------------
                          *
                          *
                          * TRANSPARENCY TEST, FORCE "O INSERTION" TO OCCUR ON ALL BYTE
                          * BOUNDRIES WITHIN MESSAGE
                          *
0408  9870 2046           TRAN-F  LDR    $R1,=A' F'
040A  9F00 11EA                   STR    $R1,<ERMG+1          REPORT TEST SECTION
                          **
                          * FORM DATA. FIRST TWO CHAR. ARE HEX 'FF'. THEN FOLLOWS A
                          * ROTATING PATTERN WITH 0'S FLOATING THROUGH A FIELD OF ONES.
                          *
040C  9B80 1800                   LAB    $B1,<SDB             SEND BUFFER ADDRESS
040E  9870 FFFF                   LDR    $R1,=Z'FFFF'
0410  9F71                        STR    $R1,+$B1             STORE FIRST TWO CHAR.
0411  9870 FC7E                   LDR    $R1,=Z'FC7E'
0413  2CC0                        LDV    $R2,=-X'40'          COUNTER
                          *
0414  9F71                TRAN1   STR    $R1,+$B1
0415  1011                        SCL    $R1,1                ROTATE PATTERN
0416  27FE                        BINC   $R2,>TRAN1
0417  AB80 0421                   LAB    $B2,<PAR6
0419  AF80 0ECF                   STB    $B2,<PARPTR
```

```
041B  8700 1174                 CL      <HEAD                   CLEAR HEADER FLAG
041D  C380 0D80                 LNJ     $B4,<GHEAD              GENERATE HEADER
                        *
                        *
041F  A380 08CD                 LNJ     $B2,<LPTS               LOOP TEST
0421  000C              PAR6    DC      X'C'                    LR 6, RCV, 8 BITS
0422  000C                      DC      X'C'                    LR7, XMIT
0423  000C                      DC      X'C'                    FILL MODE ABORT
0424  0047                      DC      =X'47'                  LR2 CONTROL, RCV, XMIT,TEST
0425  0000                      DC      X'0'                    ACTION FLAG
0426  0000                      DC      0                       RCV ACTION  FRAME, CHAR
0427  0000                      DC      0                       XMIT ACT FRAME, CHAR
0428  1800                      DC      <SDB                    DATA ADDRESS
0429  0082                      DC      X'82'                   RANGE
042A  0060                      DC      X'60'                   LAST, VALID
                        *
042B  0F00 0414                 NOP     <TRAN1
042D  8386                      JMP     $B6
                        *
                        *-----------------------------------------------------------------
                        *
                        *
                        *************
                        *    TEST G -   TESTS SUPERVISORY CONTROL BIT IN CONTROL
                        *               FIELD. THIS BIT SET SHOULD OVERIDE THE 7 BIT
                        *               CONFIGURATION GIVEN TO RECEIVE AND XMIT AND
                        *               FORCE 8 BIT MODE.
                        *
042E  9870 2047         SUP-G   LDR     $R1,=A' G'
0430  9F00 11EA                 STR     $R1,<ERMG+1             REPORT
0432  9870 8000                 LDR     $R1,=Z'8000'            SET BIT FOR SUPERVISORY FRAME
0434  9F00 1174                 STR     $R1,<HEAD
0436  AB80 0440                 LAB     $B2,<PAR7
0438  AF80 0ECF                 STB     $B2,<PARPTR
043A  B380 0D3D                 LNJ     $B3,<FACDTA             FILL ASCENDING DATA
                        *
043C  C380 0D80                 LNJ     $B4,<GHEAD              GENERATE HEADER
                        *
043E  A380 08CD                 LNJ     $B2,<LPTS               LOOP TEST
0440  0008              PAR7    DC      X'08'                   7 BITS, LR6, RCV
0441  0008                      DC      X'08'                   7 BITS, LR7, XMIT
0442  000C                      DC      X'0C'                   ABORT IDLE, 0 BIT RESIDUE
0443  0047                      DC      X'47'                   LR2, RCV, XMIT, TEST
0444  0000                      DC      X'0'                    ACTION FLAG
0445  0000                      DC      0                       RCV ACTION  FRAME, CHAR
0446  0000                      DC      0                       XMIT ACT FRAME, CHAR
0447  1800                      DC      <SDB                    DATA
0448  005A                      DC      90                      RANGE
0449  0060                      DC      X'60'                   LAST,V
                        *
044A  8386                      JMP     $B6
                        *-----------------------------------------------------------------
                        *
                        * EXERCISE LINE PROTOCOL FUNCTIONS
                        *
                        *
                        *   IF AN ERROR IS REPORTED WITH A LABEL "GX", WHERE X IS
                        *   ANY CHAR, REFER TO THE ENTRY IN TABLE "PRO-A" WHICH ENDS
                        *   WITH -X TO SEE WHAT THE TEST CONDITIONS ARE.
                        *
044B  8753              PRT-GA CL       =$R3                    SET FOR CHANNEL 0
044C  8752              PRLNLP CL       =$R2
                        *
044D  9B80 0487         PRT-LP LAB      $B1,<PRO-A              GET ADDRESS OF TABLE
044F  92ED                      LLH     $R1,$B1.+$R2            GET LR7 FOR XMIT
                        *
0450  9F00 047A         PRT-1   STR     $R1,<GLR7
0452  9570 000E                 AND     $R1,=Z'000E'            STRIP TO  BYTE SIZE
0454  9F00 047B                 STR     $R1,<GLR8
0456  1041                      SOR     $R1,1
0457  AB80 04AD                 LAB     $B2,<TCBBSZ             GET TCB CONVERSION ADD
0459  C812                      LDR     $R4,$B2.$R1
045A  CF51                      STR     $R4,=$R1
045B  1008                      SOL     $R1,8                   TCB SIZE (IF CONFIGURED)
045C  9F00 1174                 STR     $R1,<HEAD
045E  92ED                      LLH     $R1,$B1.+$R2            GET LR6 FOR RCV
045F  9F00 0479                 STR     $R1,<GLR6
0461  92ED                      LLH     $R1,$B1.+$R2            GET LCF, AFX INFO
0462  9400 1174                 OR      $R1,<HEAD
0464  9F00 1174                 STR     $R1,<HEAD
0466  92ED                      LLH     $R1,$B1.+$R2            GET TEST LABLE
0467  1982                      BNEZ    $R1,>PRT-2
0468  8386                      JMP     $B6                     DONE WITH TEST
                        *
0469  9470 4700         PRT-2   OR      $R1,=Z'4700'            PUT IN G
046B  9F00 11EA                 STR     $R1,<ERMG+1             STORE TEST LABLE
046D  AF00 077E                 STR     $R2,<DEX                STORE INDEX
                        *
                        * GENERATE DATA, ASCENDING
                        *
                        *
                        *
046F  B380 0D3D                 LNJ     $B3,<FACDTA             GENERATE DATA
0471  AB80 0479                 LAB     $B2,<GLR6
0473  AF80 0ECF                 STB     $B2,<PARPTR
0475  C380 0D80                 LNJ     $B4,<GHEAD              GENERATE HEADER
                        *
0477  A380 08CD                 LNJ     $B2,<LPTS
0479  0000              GLR6    DC      0                       LR6 FOR RCV
047A  0000              GLR7    DC      0                       LR7 FOR XMIT
047B  0000              GLR8    DC      0                       FILL MODE, ABORT IDLE
047C  0047                      DC      X'47'                   LR2,TEST,XMIT ON
047D  0000                      DC      0                       ACTION FLAG
047E  0000                      DC      0                       RCV ACTION  FRAME, CHAR
047F  0000                      DC      0                       XMIT ACT FRAME, CHAR
0480  1800                      DC      <SDB                    ADDRESS
0481  0100                      DC      X'100'                  RANGE
0482  0060                      DC      X'60'                   VALID, LAST
                        *
0483  A800 077E                 LDR     $R2,<DEX                GET BACK INDEX
0485  0F80 044D                 B       <PRT-LP
                        *
                        *
```

```
                                 *
                                 * TABLE OF TESTS FOR PROTOCAL TESTS
                                 *
                                 *   WORD 1
                                 *
                                 *        BITS 0 -7        LR7, XMIT
                                 *        BITS 8 - 15      LR6, RCV
                                 *
                                 *   WORD 2
                                 *        BITS 0 - 3       ADDRESS FIELD LENGTH
                                 *        BITS 4 - 7       LCF LENGTH
                                 *        BITS 8 - 15      ASCII TEST IDENTIFIER
                                 *
 0487   8888                     PRO-A  DC      Z'8888'              CFX, 7 BITS
 0488   0041                            DC      Z'0041'              TEST GA
 0489   2C2C                     PRO-B  DC      Z'2C2C'              TCB, 8 BITS
 048A   0042                            DC      Z'0042'              TEST GB
 048B   2828                     PRO-C  DC      Z'2828'              TCB, 7 BITS
 048C   0043                            DC      Z'0043'              TEST GC
 048D   2020                     PRO-D  DC      Z'2020'              TCB, 5 BITS
 048E   0044                            DC      Z'0044'              TEST GD
 048F   2424                     PRO-E  DC      Z'2424'              TCB, 6 BITS
 0490   0045                            DC      Z'0045'              TEST GE
 0491   4848                     PRO-F  DC      Z'4848'              ADDRESS FIELD EXTENTION 7 BITS
 0492   2046                            DC      Z'2046'              ADD = 2 BYTES
 0493   4444                     PRO-G  DC      Z'4444'              AFX, 6 BITS
 0494   3047                            DC      Z'3047'              3 BYTE ADDRESS FIELD
 0495   4040                     PRO-H  DC      Z'4040'              AFX, 5 BITS
 0496   4048                            DC      Z'4048'              4 BYT4S
 0497   4C4C                     PRO-I  DC      Z'4C4C'              AFX, 8 BITS
 0498   5049                            DC      Z'5049'              5 BYTES
 0499   0909                     PRO-J  DC      Z'0909'              LOGICAL CONTROL FIELD TEST, 7 BITS
 049A   014A                            DC      Z'014A'              1 BYTE LCF
 049B   0505                     PRO-K  DC      Z'0505'              LCF TEST, 6 BITS
 049C   024B                            DC      Z'024B'              2 BYTE LCF
 049D   0101                     PRO-L  DC      Z'0101'              LCF, 5 BITS
 049E   044C                            DC      Z'044C'              4 BYTES
 049F   0D0D                     PRO-M  DC      Z'0D0D'              LCF TEST, 8 BITS
 04A0   054D                            DC      Z'054D'              5 BYTES
 04A1   E9E9                     PRO-N  DC      Z'E9E9'              ALL FIELDS, 7 BITS
 04A2   544E                            DC      Z'544E'              5 BYTE LCF, 4 BYTE AFX
 04A3   E5E5                     PRO-O  DC      Z'E5E5'              ALL FIELDS, 6 BITS
 04A4   454F                            DC      Z'454F'              4 BYTE LCF, 5 BYTE AFX
 04A5   EDED                     PRO-P  DC      Z'EDED'              8 BITS
 04A6   1150                            DC      Z'1150'              1 LCF, 1 AFX
 04A7   E9E9                     PRO-Q  DC      Z'E9E9'
 04A8   2351                            DC      Z'2351'              2 LCF, 3 AFX
 04A9   E9C9                     PRO-R  DC      Z'E9C9'              TCB ON XMIT, NOT RECEIVE
 04AA   5552                            DC      Z'5552'
 04AB   0000                            RESV    2,0                  END OF LIST
                                 *
                                 * CONVERSION TABLE FOR BYTE SIZE CONFIG TO TCB CODE
                                 *
 04AD   0005                     TCBBSZ DC      5
 04AE   0001                            DC      1
 04AF   0006                            DC      6
 04B0   0002                            DC      2
 04B1   0007                            DC      7
 04B2   0003                            DC      3
 04B3   0000                            DC      0                    0 IS CODE FOR 8
 04B4   0004                            DC      4
 04B5   0000                            DC      0
                                 *
                                 * TRANSFER AND RECEIVE TWO MESSAGES BACK TO BACK
                                 *
 04B6   9870 2048                MULT-H LDR     $R1,=A' H'
 04B8   9F00 11EA                       STR     $R1,<ERMG+1          REPORT TEST SECTION
 04BA   9870 1100                       LDR     $R1,=Z'1100'
 04BC   9F00 117B                       STR     $R1,<EXST            EXPECTED STATUS FLAG
                                 *
                                 * FORM DATA (ASCENDING)
                                 *
 04BE   B380 0D3D                       LNJ     $B3,<FACDTA          FILL SEND BUFFER WITH ASCENDING DATA
 04C0   8700 1174                       CL      <HEAD                CLEAR HEADER FLAG
 04C2   AB80 04CA                       LAB     $B2,<PAR8
 04C4   AF80 0ECF                       STB     $B2,<PARPTR
 04C6   C380 0D80                       LNJ     $B4,<GHEAD           GENERATE HEADER
                                 *
                                 *
 04C8   A380 0A4A                       LNJ     $B2,<RET             RETURN END TEST
 04CA   0008                     PAR8   DC      8                    LR6, RCV
 04CB   0008                            DC      8                    LR7, XMIT
 04CC   0008                            DC      8                    FILL WITH ABORTS
 04CD   0047                            DC      X'47'                LR2,RCV, XMIT,TEST
 04CE   0000                            DC      X'0'                 ACTION FLAG
 04CF   0000                            DC      0                    RCV ACTION  FRAME, CHAR
 04D0   0000                            DC      0                    XMIT ACT FRAME, CHAR
                                 *
 04D1   1800                            DC      <SDB                 XMIT BUFFER 1
 04D2   0060                            DC      X'60'                RANGE
 04D3   0040                            DC      X'40'                VALID BIT
                                 *
 04D4   1830                            DC      <SDB+X'30'           DATA ADD 2
 04D5   0020                            DC      X'20'                RANGE
 04D6   0060                            DC      X'60'                LAST, VALID, NO DELAY
                                 *
 04D7   8386                            JMP     $B6
                                 *
                                 *
                                 *
                                 *---------------------------------------------------------------
                                 *
                                 *
                                 * TRANSFER, RECEIVE 2 FRAMES, FLAGS BETWEEN
                                 *
                                 *
 04D8   9870 2049                MULT-I LDR     $R1,=A' I'
 04DA   9F00 11EA                       STR     $R1,<ERMG+1          REPORT TEST SECTION
 04DC   9870 1100                       LDR     $R1,=Z'1100'         EXPECTED STATUS FLAG
 04DE   9F00 117B                       STR     $R1,<EXST
                                 *
                                 *
                                 * FORM DATA (ASCENDING)
```

```
                           *
04E0  B380 0D3D                  LNJ    $B3,<FACDTA
04E2  8700 1174                  CL     <HEAD                CLEAR HEADER FLAG
04E4  AB80 04EC                  LAB    $B2,<PAR9
04E6  AF80 0ECF                  STB    $B2,<PARPTR
04E8  C380 0D80                  LNJ    $B4,<GHEAD           GENERATE HEADER
                           *
                           *
04EA  A380 0A4A                  LNJ    $B2,<RET             RECEIVE END TEST
04EC  0004                 PAR9  DC     4                    LR6, RCV
04ED  0004                       DC     4                    LR7 , XMIT
04EE  0084                       DC     X'84'                FILL WITH FLAGS
04EF  0047                       DC     =X'47'               LR2 CONTROL, RCV, XMIT, TEST
04F0  0020                       DC     X'20'                ACTION FLAG, DELAY BETWEEN FRAMES
04F1  0000                       DC     0                    RCV ACTION  FRAME, CHAR
04F2  0000                       DC     0                    XMIT ACT FRAME, CHAR
04F3  1800                       DC     <SDB                 SEND BUFFER
04F4  0010                       DC     X'10'                RANGE
04F5  0040                       DC     X'40'                VALID
                           *
04F6  1808                       DC     <SDB+X'8'            ADD, 2ND CCB
04F7  0010                       DC     X'10'                RANGE
04F8  0060                       DC     X'60'                LAST,VALID,
                           *
04F9  0F5F                       NOP    >MULT-I
                           *
04FA  8386                       JMP    $B6                  EXIT TEST
                           *------------------------------------------------------------------
                           *
                           *
                           * TRANSFER, RECEIVE 2 FRAMES, FLAGS BETWEEN, DELAYED XMIT OF XMIT FIFO
                           *
                           *
04FB  9870 4941            MUL-IA LDR   $R1,=A'IA'
04FD  9F00 11EA                  STR    $R1,<ERMG+1          REPORT TEST SECTION
04FF  9870 1100                  LDR    $R1,=Z'1100'         EXPECTED STATUS FLAG
0501  9F00 117B                  STR    $R1,<EXST
                           *
                           *
                           * FORM DATA (ASCENDING)
                           *
0503  B380 0D3D                  LNJ    $B3,<FACDTA
0505  8700 1174                  CL     <HEAD                CLEAR HEADER FLAG
0507  AB80 050F                  LAB    $B2,<PAR9IA
0509  AF80 0ECF                  STB    $B2,<PARPTR
050B  C380 0D80                  LNJ    $B4,<GHEAD           GENERATE HEADER
                           *
                           *
050D  A380 0A4A                  LNJ    $B2,<RET             RECEIVE END TEST
050F  0004                 PAR9IA DC    4                    LR6, RCV
0510  0004                       DC     4                    LR7 , XMIT
0511  00C4                       DC     X'C4'                FILL WITH FLAGS , DELAYED XMIT
0512  0047                       DC     =X'47'               LR2 CONTROL, RCV, XMIT, TEST
0513  0020                       DC     X'20'                ACTION FLAG, DELAY BETWEEN FRAMES
0514  0000                       DC     0                    RCV ACTION  FRAME, CHAR
0515  0000                       DC     0                    XMIT ACT FRAME, CHAR
0516  1800                       DC     <SDB                 SEND BUFFER
0517  0010                       DC     X'10'                RANGE
0518  0040                       DC     X'40'                VALID
                           *
0519  1808                       DC     <SDB+X'8'            ADD, 2ND CCB
051A  0010                       DC     X'10'                RANGE
051B  0060                       DC     X'60'                LAST,VALID,
                           *
051C  0F5F                       NOP    >MUL-IA
                           *
051D  8386                       JMP    $B6                  EXIT TEST
                           *
                           *------------------------------------------------------------------
                           *
                           * SINGLE FRAME TERMINATION TESTS
                           *
                           *
                           *    IN THIS TEST FRAMES ARE ABORTED IN VARIOUS STATES WITHIN
                           *   THE ADDRESS, CONTROL, LCF, AND TEXT FIELDS.
                           *
                           *   IF AN ERROR IS REPORTED WITH A LABLE "JX" WHERE X IS ANY
                           *   CHAR, REFER TO THE ENTRY IN TABLE "TRMTBL" WHICH ENDS WITH
                           *   -X TO FIND THE TEST CONDITIONS.
                           *
051E  8752                 TERM-J CL    =$R2                 INDEX
                           *
051F  9B80 056C            TRMMLP LAB   $B1,<TRMTBL          TABLE OF TESTS
0521  92ED                       LLH    $R1,$B1.+$R2         GET XMIT LR7
0522  9F00 055E                  STR    $R1,<TRML7
0524  9570 000E                  AND    $R1,=Z'000E'         STRIP TO BYTE SIZE
0526  9F00 055F                  STR    $R1,<TRML8           STORE FOR LAST BYTE
0528  1041                       SOR    $R1,1
                           * CONVERT FROM MLCP BYTE SIZE CODE TO TCB CODE
0529  C810 04AD                  LDR    $R4,<TCBBSZ.$R1
052B  CF51                       STR    $R4,=$R1
                           *
052C  1008                       SOL    $R1,8
052D  9F00 1174                  STR    $R1,<HEAD            TCB INFO
052F  92ED                       LLH    $R1,$B1.+$R2         GET RCV LR6
0530  9F00 055D                  STR    $R1,<TRML6
0532  92ED                       LLH    $R1,$B1.+$R2         GET LCF, AFX INFO
0533  9400 1174                  OR     $R1,<HEAD
0535  9F00 1174                  STR    $R1,<HEAD
0537  92ED                       LLH    $R1,$B1.+$R2         GET TEST LABLE
0538  1900 056B                  BEZ    $R1,<END-J
053A  9470 4A00                  OR     $R1,=Z'4A00'         FORM TEST J-
053C  9F00 11EA                  STR    $R1,<ERMG+1
                           *
053E  92ED                       LLH    $R1,$B1.+$R2         FLAG
053F  9470 0080                  OR     $R1,=X'80'           PUT IN BIT FOR CCP TO COUNT CHAR
0541  9F00 0561                  STR    $R1,<TRMFLG
0543  92ED                       LLH    $R1,$B1.+$R2         XMIT CHAR COUNT
0544  AF00 077E                  STR    $R2,<DEX             STORE INDEX
0546  9F52                       STR    $R1,=$R2
0547  9470 FF00                  OR     $R1,=Z'FF00'
0549  9F00 0563                  STR    $R1,<TRMAC           FRAME
054B  8252                       NEG    =$R2
```

```
054C  8AD2                    INC    =$R2                  SET RANGE = ABORT COUNT + 1
054D  A570 00FF               AND    $R2,=Z'00FF'
                      *
054F  0F04                    NOP    >$+4
0550  0F04                    NOP    >$+4
0551  AF00 0565               STR    $R2,<TRMRNG           RANGE
                      *
                      * GENERATE DATA, ASCENDING
                      *
0553  B380 0D3D               LNJ    $B3,<FACDTA
                      *
0555  AB80 055D               LAB    $B2,<TRML6
0557  AF80 0ECF               STB    $B2,<PARPTR
0559  C380 0D80               LNJ    $B4,<GHEAD            GENERATE HEADER
                      *
055B  A380 0970               LNJ    $B2,<ABUND            ABORT, UNDERUN TEST
055D  0000            TRML6   DC     0                     LR6 FOR RCV
055E  0000            TRML7   DC     0                     LR7 FOR XMIT
055F  0000            TRML8   DC     0                     FILL MODE, ABORT IDLE
0560  0047                    DC     X'47'                 LR2,TEST, XMIT ON
0561  0000            TRMFLG  DC     0                     ACTION FLAG
0562  0000                    DC     0                     RCV ACTION CHAR, FRAME
0563  0000            TRMAC   DC     0                     XMIT FRAME, CHAR
0564  1800                    DC     <SDB                  ADDRESS
0565  0200            TRMRNG  DC     X'200'                RANGE
0566  0060                    DC     X'60'                 VALID, LAST
                      *
0567  A800 077E               LDR    $R2,<DEX              RESTORE INDEX
0569  0F80 051F               B      <TRMMLP               DO NEXT TEST
                      *
056B  8386            END-J   JMP    $B6
                      *
                      * TABLE FOR TERMINATION TESTS
                      *
                      * WORD 1
                      *   BITS 0-7  LR7, XMIT
                      *   BITS 8-15  LR6, RCV
                      *
                      * WORD 2
                      *   BITS 0 - 3     ADDRESS FIELD LENGTH
                      *        4 - 7     LCF LENGTH
                      *        7 - 15    ASCII TEST LABLE
                      *
                      *
                      * WORD 3
                      *
                      *   BITS 0 - 7     ACTION FLAG
                      *   BITS 8 - 15    ACTION CHAR
                      *
                      *   ALL TESTS ARE IN 8 BIT MODE
                      *
                      *
                      * ABORT DURING ADDRESS STATE
                      *
056C  0C0C            TRMTBL  DC     Z'0C0C'              8 BIT CHAR
056D  0041                    DC     Z'0041'              TEST JA
056E  02FD                    DC     Z'02FD'              ABORT, 3 CHAR
                      *
                      *  ABORT DURING ADDRESS FIELD EXTENTION
                      *
056F  0C0C            TRM-B   DC     Z'0C0C'              AFX, 8 BITS
0570  1042                    DC     Z'1042'              TEST JB
0571  02FC                    DC     Z'02FC'              ABORT, 4 CHAR
                      *
                      * ABORT DURING ADDRESS FIELD EXTENTION, > 32 BITS
                      *
0572  4C4C            TRM-C   DC     Z'4C4C'              AFX, 8 BITS
0573  3043                    DC     Z'3043'              3 BYTE AF, TEST JC
0574  02FB                    DC     Z'02FB'              ABORT, 5 CHAR
                      *
                      * ABORT DURING CONTROL FIELD STATE
                      *
0575  0C0C            TRM-D   DC     Z'0C0C'              8 BITS, CFX
0576  0044                    DC     =Z'0044'             TEST JD
0577  02FC                    DC     =Z'02FC'             ABORT, 4 CHAR
                      *
                      * ABORT DURING CONTROL FIELD STATE, > 32 BITS
                      *
0578  4C4C            TRM-E   DC     Z'4C4C'              CF, AFX
0579  3045                    DC     =Z'3045'             TEST JE
057A  02FA                    DC     Z'02FA'              ABORT, 6 CHAR
                      *
                      * ABORT DURING EXTENDED CONTROL FIELD STATE
                      *
057B  8C8C            TRM-F   DC     Z'8C8C'              8 BITS, CFX
057C  0046                    DC     Z'0046'              TEST JF
057D  02FB                    DC     =Z'02FB'             ABORT, 5 CHAR
                      *
                      * ABORT DURING TCB STATE
                      *
057E  2C2C            TRM-G   DC     Z'2C2C'              TCB
057F  0047                    DC     Z'0047'              TEST JG
0580  02FB                    DC     Z'02FB'              ABORT, 5 CHAR
                      *
                      * ABORT DURING LCF STATE
                      *
0581  0D0D            TRM-H   DC     Z'0D0D'              LCF
0582  0448                    DC     Z'0448'              4 BYTE LCF  TEST JH
0583  02FB                    DC     Z'02FB'              ABORT, 5 CHAR
                      *
                      * ABORT DURING TEXT STATE
                      *
0584  0C0C            TRM-J   DC     Z'0C0C'              8 BITS
0585  004A                    DC     Z'004A'              TEST JJ
0586  02FB                    DC     Z'02FB'              ABORT, 5 CHAR
0587  0000                    RESV   12,0                 END
                      *-----------------------------------------------------------------
                      *
                      *
                      * TRANSFER, RECEIVE 2 FRAMES, ABORTS BETWEEN
                      *
                      *
0593  9870 204C       MULT-L  LDR    $R1,=A' L'
```

```
0595  9F00 11EA              STR    $R1,<ERMG+1         REPORT TEST SECTION
0597  9870 1100              LDR    $R1,=Z'1100'        TURN ON RCV AFTER 1 XMIT CHAR
0599  9F00 117B              STR    $R1,<EXST
                      *
                      *
                      * FORM DATA (ASCENDING)
                      *
059B  B380 0D3D              LNJ    $B3,<FACDTA
059D  8700 1174              CL     <HEAD               CLEAR HEADER FLAG
059F  AB80 05A7              LAB    $B2,<PAR10
05A1  AF80 0ECF              STB    $B2,<PARPTR
05A3  C380 0D80              LNJ    $B4,<GHEAD          GENERATE HEADER
                      *
                      *
05A5  A380 0A4A              LNJ    $B2,<RET            RECEIVE END TEST
05A7  0004            PAR10  DC     4                   LR6, RCV
05A8  0004                   DC     4                   LR7, XMIT
05A9  0004                   DC     4                   FILL WITH ABORTS
05AA  0047                   DC     =X'47'              LR2 CONTROL, RCV, XMIT, TEST
05AB  0020                   DC     X'20'               ACTION FLAG, DELAY BETWEEN FRAMES
05AC  0000                   DC     0                   RCV ACT FRAME, CHAR
05AD  0000                   DC     =X'0'               XMIT ACT FRAME, CHAR
                      *
05AE  1800                   DC     <SDB                SEND BUFFER
05AF  0010                   DC     X'10'               RANGE
05B0  0040                   DC     X'40'               VALID
                      *
05B1  1808                   DC     <SDB+X'8'           ADD, 2ND CCB
05B2  0010                   DC     X'10'               RANGE
05B3  0060                   DC     X'60'               LAST, VALID
                      *
05B4  0F04                   NOP    >$+4
                      *
05B5  8386                   JMP    $B6                 EXIT TEST
                      *
                      *-----------------------------------------------------------
                      *
                      *
                      * UNDERUN TEST 1 - DELAY 100 MS AFTER TRANSMITTING 5 CHAR
                      *
05B6  9870 204B       UND1-K LDR    $R1,=A' K'
05B8  9F00 11EA              STR    $R1,<ERMG+1         REPORT TEST SECTION
                      * FORM ASCENDING DATA
05BA  B380 0D3D              LNJ    $B3,<FACDTA
05BC  8700 1174              CL     <HEAD               CLEAR HEADER FLAG
05BE  AB80 05C6              LAB    $B2,<PAR11
05C0  AF80 0ECF              STB    $B2,<PARPTR
05C2  C380 0D80              LNJ    $B4,<GHEAD          GENERATE HEADER
                      *
                      *
05C4  A380 0970              LNJ    $B2,<ABUND          ABORT, UNDERUN TEST
05C6  0008            PAR11  DC     =X'8'               RCV CONFIG, 7 BITS
05C7  0008                   DC     =X'8'               XMIT CONFIG, 7 BITS
05C8  0008                   DC     =8                  FILL MODE, ABORTS
05C9  0047                   DC     =X'47'              LR2, RCV, XMIT, TEST ON
05CA  0081                   DC     X'81'               ACTION FLAG DELAY  INSERT ON XMIT
05CB  0000                   DC     0                   RC ACT FR, CHR
05CC  FFFB                   DC     =Z'FFFB'            DELAY FR 1, CHR 5
05CD  1800                   DC     <SDB                SEND BUFFER
05CE  0007                   DC     7                   RANGE
05CF  0060                   DC     X'60'               LAST , VALID
                      *
05D0  8386                   JMP    $B6
                      *-----------------------------------------------------------
                      *
                      *   SINGLE FRAME TESTS
                      *
                      *        THE FOLLOWING TESTS (EA - ED) ARE DESIGNED TO TEST
                      *        ARE DESIGNED TO TEST DIFFERENT END CONDITIONS OF
                      *        SINGLE FRAME MESSAGES FILLING AND OVERFLOWING THE
                      *        RCV FIFO BUFFER. TO DO THIS THE RECEIVER IS NOT
                      *        TURNED ON UNTIL AFTER THE FIRST XMIT FRAME IS
                      *        TRANSMIT.
                      *
                      *-----------------------------------------------------------
05D1  9870 4541       RCV-EA LDR    $R1,=A'EA'
05D3  9F00 11EA              STR    $R1,<ERMG+1         REPORT TEST
05D5  9870 1000              LDR    $R1,=Z'1000'        EXPECTED STATUS
05D7  9F00 117B              STR    $R1,<EXST
                      *
05D9  B380 0D3D              LNJ    $B3,<FACDTA         FILL ASCENDING DATA
05DB  8700 1174              CL     <HEAD               CLEAR HEADER FLAG
05DD  AB80 05E5              LAB    $B2,<PAR12
05DF  AF80 0ECF              STB    $B2,<PARPTR
05E1  C380 0D80              LNJ    $B4,<GHEAD          GENERATE HEADER
                      *
                      *
05E3  A380 0A4A              LNJ    $B2,<RET            RECEIVE END TEST
05E5  000C            PAR12  DC     =X'C'               RCV CONFIGURATION, 8 BITS
05E6  000C                   DC     =X'C'               XMIT CONFIG,  8 BITS
05E7  008C                   DC     X'8C'               FILL WITH FLAGS
05E8  0045                   DC     =X'45'              LR2 CONTROL, TEST, XMIT
05E9  0084                   DC     X'84'               FLAG, DELAYED TURN ON OF RCV
05EA  0000                   DC     0
05EB  FF00                   DC     Z'FF00'             XMIT CONTROL FRAME, CHR
                      *
05EC  1800                   DC     <SDB                SEND BUFFER
05ED  0040                   DC     64                  RANGE IN BYTES
05EE  0060                   DC     X'60'               VALID
                      *
05EF  8386                   JMP    $B6
                      *-----------------------------------------------------------
05F0  9870 4542       RCV-EB LDR    $R1,=A'EB'
05F2  9F00 11EA              STR    $R1,<ERMG+1         REPORT TEST
05F4  9870 1000              LDR    $R1,=Z'1000'        EXPECTED STATUS
05F6  9F00 117B              STR    $R1,<EXST
                      *
05F8  B380 0D3D              LNJ    $B3,<FACDTA         FILL ASCENDING DATA
05FA  8700 1174              CL     <HEAD               CLEAR HEADER FLAG
05FC  AB80 0604              LAB    $B2,<PAR13
05FE  AF80 0ECF              STB    $B2,<PARPTR
0600  C380 0D80              LNJ    $B4,<GHEAD          GENERATE HEADER
                      *
```

```
                          *
0602  A380 0A4A                 LNJ    $B2,<RET              RECEIVE END TEST
0604  000C                PAR13 DC     =X'C'                 RCV CONFIGURATION, 8 BITS
0605  000C                      DC     =X'C'                 XMIT CONFIG,  8 BITS
0606  008C                      DC     X'8C'                 FILL WITH FLAGS
0607  0045                      DC     =X'45'                LR2 CONTROL, TEST, XMIT
0608  0084                      DC     X'84'                 FLAG, DELAYED TURN ON OF RCV
0609  0000                      DC     0
060A  FF00                      DC     Z'FF00'               XMIT CONTROL FRAME, CHR
                          *
060B  1800                      DC     <SDB                  SEND BUFFER
060C  0041                      DC     65                    RANGE IN BYTES
060D  0060                      DC     X'60'                 VALID
                          *
060E  8386                      JMP    $B6
                          *------------------------------------------------------------------
060F  9870 4543           RCV-EC LDR   $R1,=A'EC'
0611  9F00 11EA                 STR    $R1,<ERMG+1           REPORT TEST
0613  9870 1000                 LDR    $R1,=Z'1000'          EXPECTED STATUS
0615  9F00 117B                 STR    $R1,<EXST
                          *
0617  B380 0D3D                 LNJ    $B3,<FACDTA           FILL ASCENDING DATA
0619  8700 1174                 CL     <HEAD                 CLEAR HEADER FLAG
061B  AB80 0623                 LAB    $B2,<PAR14
061D  AF80 0ECF                 STB    $B2,<PARPTR
061F  C380 0D80                 LNJ    $B4,<GHEAD            GENERATE HEADER
                          *
                          *
0621  A380 0A4A                 LNJ    $B2,<RET              RECEIVE END TEST
0623  000C                PAR14 DC     =X'C'                 RCV CONFIGURATION, 8 BITS
0624  000C                      DC     =X'C'                 XMIT CONFIG,  8 BITS
0625  008C                      DC     X'8C'                 FILL WITH FLAGS
0626  0045                      DC     =X'45'                LR2 CONTROL, TEST, XMIT
0627  0084                      DC     X'84'                 FLAG, DELAYED TURN ON OF RCV
0628  0000                      DC     0
0629  FF00                      DC     Z'FF00'               XMIT CONTROL FRAME, CHR
                          *
062A  1800                      DC     <SDB                  SEND BUFFER
062B  0042                      DC     66                    RANGE IN BYTES
062C  0060                      DC     X'60'                 VALID
                          *
062D  8386                      JMP    $B6
                          *------------------------------------------------------------------
062E  9870 4544           RCV-ED LDR   $R1,=A'ED'
0630  9F00 11EA                 STR    $R1,<ERMG+1           REPORT TEST
0632  9870 2000                 LDR    $R1,=Z'2000'          EXPECTED STATUS FLAG
0634  9F00 117B                 STR    $R1,<EXST
                          *
0636  B380 0D3D                 LNJ    $B3,<FACDTA           FILL ASCENDING DATA
0638  8700 1174                 CL     <HEAD                 CLEAR HEADER FLAG
063A  AB80 0642                 LAB    $B2,<PAR15
063C  AF80 0ECF                 STB    $B2,<PARPTR
063E  C380 0D80                 LNJ    $B4,<GHEAD            GENERATE HEADER
                          *
                          *
0640  A380 0A4A                 LNJ    $B2,<RET              RECEIVE END TEST
0642  000C                PAR15 DC     =X'C'                 RCV CONFIGURATION, 8 BITS
0643  000C                      DC     =X'C'                 XMIT CONFIG,  8 BITS
0644  000C                      DC     X'C'                  FILL WITH ABORTS
0645  0045                      DC     =X'45'                LR2 CONTROL, TEST, XMIT
0646  0084                      DC     X'84'                 FLAG, DELAYED TURN ON OF RCV
0647  0000                      DC     0                     RCV CONTROL FRAME, CHAR
0648  FF00                      DC     Z'FF00'               XMIT FRAME CONTROL INFO
                          *
0649  1800                      DC     <SDB                  SEND BUFFER
064A  0043                      DC     67                    RANGE IN BYTES
064B  0060                      DC     X'60'                 VALID
                          *
064C  8386                      JMP    $B6
                          *------------------------------------------------------------------
                          *------------------------------------------------------------------
                          *
                          * TWO FRAME TESTS
                          *
                          *   THE FOLLOWING TESTS TRANSMITS 2 FRAMES BEFORE TURNING ON RCV.
                          *   THEY ARE DESIGNED TO TEST VARIOUS END CONDITIONS. REFER
                          *   TO TABLE "PAR" AND FIND THE LABLE GIVEN IN THE ERROR
                          *   PRINTOUT TO FIND SPECIFIC TEST CONDITIONS.
                          *
                          *   REFER TO THE DESCRIPTION OF THE CCP CONTROL FLAG IN THE
                          *   HEADER FOR ACTION TAKEN BY EACH TEST.
                          *
                          * RCV IS NOT TURNED ON UNTIL AFTER BOTH FRAMES ARE TRANSMITTED
                          * FOR MOST OF THESE TESTS.
                          *
064D  8751                TFR-M CL     =$R1
064E  9B80 069A           TLUP  LAB    $B1,<PAR
                          *
0650  A85D                      LDR    $R2,$B1.+$R1          GET LABLE
0651  2900 0699                 BEZ    $R2,<ETFR             B = DONE
0653  AF00 11EA                 STR    $R2,<ERMG+1
0655  A85D                      LDR    $R2,$B1.+$R1          GET EXPECTED STATUS FLAG
0656  AF00 117B                 STR    $R2,<EXST
0658  A85D                      LDR    $R2,$B1.+$R1          GET CCP CONTROL FLAG
0659  AF00 068C                 STR    $R2,<TFLG
065B  A85D                      LDR    $R2,$B1.+$R1          GET RANGE OF FIRST
065C  AF00 0690                 STR    $R2,<TRNG1
065E  A85D                      LDR    $R2,$B1.+$R1          GET RANGE 2
065F  AF00 0693                 STR    $R2,<TRNG2
0661  A85D                      LDR    $R2,$B1.+$R1          GET XMIT CONTROL FRAME, CHAR (NEG)
0662  AF00 068E                 STR    $R2,<XCON
0664  9F00 077E                 STR    $R1,<DEX              STORE INDEX
                          *
                          *
                          *
                          * IF ONE BYTE + CRC FOLLOWED BY FLAG, CHANGE TO 3 BYTES
                          * + NO CRC
                          *
0666  1C0C                      LDV    $R1,=X'C'
0667  9F00 0689                 STR    $R1,<PAR22            SET FOR 8 BIT INITIAL
0669  8280 068C                 LB     <TFLG,=Z'0002'        GET ABORT BIT
066B  0002
066C  0510                      BBT    >MTS1                 B = ABORT
```

```
066D  A800 0693                LDR     $R2,<TRNG2            RANGE 2
066F  2D01                     CMV     $R2,=1
0670  098C                     BNE     >MTS1
                        *
0671  A870 FEFD                LDR     $R2,=Z'FEFD'
0673  AF00 068E                STR     $R2,<XCON
0675  2C03                     LDV     $R2,=3
0676  AF00 0693                STR     $R2,<TRNG2
0678  9870 002E                LDR     $R1,=X'2E'
067A  9F00 0689                STR     $R1,<PAR22            SET FOR NO CRC
                        *
067C  B380 0D3D         MTS1   LNJ     $B3,<FACDTA           FILL ASCENDING DATA
067E  8700 1174                CL      <HEAD                 CLEAR HEADER FLAG
0680  AB80 0688                LAB     $B2,<PAR16
0682  AF80 0ECF                STB     $B2,<PARPTR
0684  C380 0D80                LNJ     $B4,<GHEAD            GENERATE HEADER
                        *
                        *
0686  A380 0A4A                LNJ     $B2,<RET              RECEIVE END TEST
0688  000C              PAR16  DC      X'C'                  8 BYTE, RCV
0689  000C              PAR22  DC      X'C'                  8 BYTE, XMIT
068A  008C                     DC      X'8C'                 FLAG IDLE BETWEEN
068B  0045                     DC      X'45'                 XMIT, TEST (LR 2)
068C  0000              TFLG   DC      0                     CONTROL FLAG
068D  0000                     DC      0                     RCV CONTROL, FRAME, CHAR
068E  FE00              XCON   DC      Z'FE00'               XMIT CONTROL FRAME, CHAR
                        *
068F  1800                     DC      <SDB                  SEND BUFFER 1
0690  0000              TRNG1  DC      0
0691  0040                     DC      X'40'                 CONTROL
                        *
0692  1840                     DC      <SDB+X'40'            SEND BUFFER 2
0693  0000              TRNG2  DC      0                     RANGE 2
0694  0060                     DC      X'60'                 CONTROL
                        *
0695  9800 077E                LDR     $R1,<DEX
0697  0F80 064E                B       <TLUP                 DO NEXT 2 FRAME TEST
                        *
0699  8386              ETFR   JMP     $B6                   END OF TEST
                        *
                        * TABLE FOR TWO   FRAME TESTS
                        *
069A  6D61              PAR    DC      =A'MA'                LABEL
069B  1100                     DC      X'1100'               EXPECTED STATUS FLAG
069C  0084                     DC      X'84'                 CCP CONTROL FLAG
069D  003F                     DC      63                    RANGE, 1ST FRAME
069E  0002                     DC      X'2'                  RANGE , 2ND FRAME
069F  FEFE                     DC      Z'FEFE'               XMIT CONTROL, FRAME 2, CHAR 2
                        *
06A0  6D62                     DC      =A'MB'                LABEL
06A1  1300                     DC      X'1300'               EXPECTED STATUS FLAG
06A2  0084                     DC      X'84'                 CCP CONTROL FLAG
06A3  003F                     DC      63                    RANGE, 1ST FRAME
06A4  0001                     DC      1                     RANGE OF SECOND
06A5  FEFE                     DC      Z'FEFE'               XMIT CONTROL, FRAME 2, CHAR 2
                        *
06A6  6D63                     DC      =A'MC'                LABEL
06A7  1100                     DC      X'1100'               EXPECTED STATUS FLAG
06A8  0084                     DC      X'84'                 CCP CONTROL FLAG
06A9  003F                     DC      63                    RANGE, 1ST FRAME
06AA  0003                     DC      3                     RANGE, 2ND FRAME
06AB  FEFD                     DC      Z'FEFD'               XMIT CONTROL, FRAME 2, CHAR 3
                        *
06AC  6D64                     DC      =A'MD'                LABEL
06AD  1200                     DC      X'1200'               EXPECTED STATUS FLAG
06AE  0084                     DC      X'84'                 CCP CONTROL FLAG
06AF  003F                     DC      63                    RANGE, 1ST FRAME
06B0  0004                     DC      4                     RANGE, 2ND FRAME
06B1  FEFC                     DC      Z'FEFC'               XMIT CONTROL, FRAME 2, CHAR 4
                        *
06B2  6D65                     DC      =A'ME'                LABEL
06B3  1300                     DC      X'1300'               EXPECTED STATUS FLAG
06B4  0086                     DC      X'86'                 CCP CONTROL FLAG
06B5  003F                     DC      63                    RANGE, 1ST FRAME
06B6  0003                     DC      3                     RANGE, 2ND FRAME
06B7  FEFE                     DC      Z'FEFE'               XMIT CONTROL, FRAME 2, CHAR 2
                        *
06B8  6D66                     DC      =A'MF'                LABEL
06B9  1300                     DC      X'1300'               EXPECTED STATUS FLAG
06BA  0086                     DC      X'86'                 CCP CONTROL FLAG
06BB  003F                     DC      63                    RANGE, 1ST FRAME
06BC  0004                     DC      4                     RANGE, 2ND FRAME
06BD  FEFD                     DC      Z'FEFD'               XMIT CONTROL, FRAME 2, CHAR 3
                        *
06BE  6D67                     DC      =A'MG'                LABEL
06BF  1600                     DC      X'1600'               EXPECTED STATUS FLAG
06C0  0086                     DC      X'86'                 CCP CONTROL FLAG
06C1  003F                     DC      63                    RANGE, 1ST FRAME
06C2  0006                     DC      6                     RANGE, 2ND FRAME
06C3  FEFB                     DC      Z'FEFB'               XMIT CONTROL, FRAME 2, CHAR 5
                        *
                        *
06C4  6D2A                     DC      =A'M*'
06C5  1700                     DC      X'1700'               EXPECTED STAUS FLAG
06C6  0086                     DC      X'86'                 CCP FLAG
06C7  003F                     DC      63                    RANGE 1
06C8  0007                     DC      7                     RANGE 2
06C9  FEFA                     DC      Z'FEFA'               XMIT CONTROL FRAME 2, CHAR 6
                        *
06CA  6D68                     DC      =A'MH'                LABEL
06CB  1100                     DC      X'1100'               EXPECTED STATUS FLAG
06CC  0084                     DC      X'84'                 CCP CONTROL FLAG
06CD  0040                     DC      64                    RANGE, 1ST FRAME
06CE  0002                     DC      2                     RANGE, 2ND FRAME
06CF  FEFE                     DC      Z'FEFE'               XMIT CONTROL, FRAME 2, CHAR 2
                        *
06D0  6D69                     DC      =A'MI'                LABEL
06D1  1300                     DC      X'1300'               EXPECTED STATUS FLAG
06D2  0084                     DC      X'84'                 CCP CONTROL FLAG
06D3  0040                     DC      64                    RANGE, 1ST FRAME
06D4  0001                     DC      1                     RANGE, 2ND FRAME
06D5  FEFF                     DC      Z'FEFF'               XMIT CONTROL, FRAME 2, CHAR 1
                        *
```

```
06D6   6D6A              DC      =A'MJ'          LABEL
06D7   1200              DC      X'1200'         EXPECTED STATUS FLAG
06D8   0084              DC      X'84'           CCP CONTROL FLAG
06D9   0040              DC      64              RANGE, 1ST FRAME
06DA   0003              DC      3               RANGE, 2ND FRAME
06DB   FEFD              DC      Z'FEFD'         XMIT CONTROL, FRAME 2, CHAR 3
                 *
06DC   6D6B              DC      =A'MK'          LABEL
06DD   1200              DC      X'1200'         EXPECTED STATUS FLAG
06DE   0084              DC      X'84'           CCP CONTROL FLAG
06DF   0040              DC      64              RANGE, 1ST FRAME
06E0   0004              DC      4               RANGE, 2ND FRAME
06E1   FEFC              DC      Z'FEFC'         XMIT CONTROL, FRAME 2, CHAR 4
                 *
06E2   6D6C              DC      =A'ML'          LABEL
06E3   1300              DC      X'1300'         EXPECTED STATUS FLAG
06E4   0086              DC      X'86'           CCP CONTROL FLAG
06E5   0040              DC      64              RANGE, 1ST FRAME
06E6   0003              DC      3               RANGE, 2ND FRAME
06E7   FEFE              DC      Z'FEFE'         XMIT CONTROL, FRAME 2, CHAR 2
                 *
06E8   6D6D              DC      =A'MM'          LABEL
06E9   1300              DC      X'1300'         EXPECTED STATUS FLAG
06EA   0086              DC      X'86'           CCP CONTROL FLAG
06EB   0040              DC      64              RANGE, 1ST FRAME
06EC   0004              DC      4               RANGE, 2ND FRAME
06ED   FEFD              DC      Z'FEFD'         XMIT CONTROL, FRAME 2, CHAR 3
                 *
06EE   6D6E              DC      =A'MN'          LABEL
06EF   1300              DC      X'1300'         EXPECTED STATUS FLAG
06F0   0086              DC      X'86'           CCP CONTROL FLAG
06F1   0040              DC      64              RANGE, 1ST FRAME
06F2   0005              DC      5               RANGE, 2ND FRAME
06F3   FEFC              DC      Z'FEFC'         XMIT CONTROL, FRAME 2, CHAR 4
                 *
06F4   6D6F              DC      =A'MO'          LABEL
06F5   1300              DC      X'1300'         EXPECTED STATUS FLAG
06F6   0084              DC      X'84'           CCP CONTROL FLAG
06F7   0041              DC      65              RANGE, 1ST FRAME
06F8   0001              DC      1               RANGE, 2ND FRAME
06F9   FEFF              DC      Z'FEFF'         XMIT CONTROL, FRAME 2, CHAR 1
                 *
06FA   6D70              DC      =A'MP'          LABEL
06FB   1500              DC      X'1500'         EXPECTED STATUS FLAG
06FC   0084              DC      X'84'           CCP CONTROL FLAG
06FD   0041              DC      65              RANGE, 1ST FRAME
06FE   0002              DC      2               RANGE, 2ND FRAME
06FF   FEFE              DC      Z'FEFE'         XMIT CONTROL, FRAME 2, CHAR 2
                 *
0700   6D71              DC      =A'MQ'          LABEL
0701   1500              DC      X'1500'         EXPECTED STATUS FLAG
0702   0084              DC      X'84'           CCP CONTROL FLAG
0703   0041              DC      65              RANGE, 1ST FRAME
0704   0003              DC      3               RANGE, 2ND FRAME
0705   FEFD              DC      Z'FEFD'         XMIT CONTROL, FRAME 2, CHAR 3
                 *
0706   6D72              DC      =A'MR'          LABEL
0707   1500              DC      X'1500'         EXPECTED STATUS FLAG
0708   0084              DC      X'84'           CCP CONTROL FLAG
0709   0041              DC      65              RANGE, 1ST FRAME
070A   0004              DC      4               RANGE, 2ND FRAME
070B   FEFC              DC      Z'FEFC'         XMIT CONTROL, FRAME 2, CHAR 4
                 *
070C   6D73              DC      =A'MS'          LABEL
070D   1300              DC      X'1300'         EXPECTED STATUS FLAG
070E   0086              DC      X'86'           CCP CONTROL FLAG
070F   0041              DC      65              RANGE, 1ST FRAME
0710   0003              DC      3               RANGE, 2ND FRAME
0711   FEFE              DC      Z'FEFE'         XMIT CONTROL, FRAME 2, CHAR 2
                 *
0712   6D74              DC      =A'MT'          LABEL
0713   1300              DC      X'1300'         EXPECTED STATUS FLAG
0714   0086              DC      X'86'           CCP CONTROL FLAG
0715   0041              DC      65              RANGE, 1ST FRAME
0716   0004              DC      4               RANGE, 2ND FRAME
0717   FEFD              DC      Z'FEFD'         XMIT CONTROL, FRAME 2, CHAR 3
                 *
0718   6D75              DC      =A'MU'          LABEL
0719   1300              DC      X'1300'         EXPECTED STATUS FLAG
071A   0086              DC      X'86'           CCP CONTROL FLAG
071B   0041              DC      65              RANGE, 1ST FRAME
071C   0005              DC      5               RANGE, 2ND FRAME
071D   FEFC              DC      Z'FEFC'         XMIT CONTROL, FRAME 2, CHAR 4
                 *
071E   6D76              DC      =A'MV'          LABEL
071F   1300              DC      X'1300'         EXPECTED STATUS FLAG
0720   0084              DC      X'84'           CCP CONTROL FLAG
0721   0042              DC      66              RANGE, 1ST FRAME
0722   0001              DC      1               RANGE, 2ND FRAME
0723   FEFF              DC      Z'FEFF'         XMIT CONTROL, FRAME 2, CHAR 1
                 *
                 *
0724   6D77              DC      =A'MW'          LABEL
0725   1500              DC      X'1500'         EXPECTED STATUS FLAG
0726   0084              DC      X'84'           CCP CONTROL FLAG
0727   0042              DC      66              RANGE, 1ST FRAME
0728   0002              DC      2               RANGE, 2ND FRAME
0729   FEFE              DC      Z'FEFE'         XMIT CONTROL, FRAME 2, CHAR 2
                 *
072A   6D78              DC      =A'MX'          LABEL
072B   1500              DC      X'1500'         EXPECTED STATUS FLAG
072C   0084              DC      X'84'           CCP CONTROL FLAG
072D   0042              DC      66              RANGE, 1ST FRAME
072E   0003              DC      3               RANGE, 2ND FRAME
072F   FEFD              DC      Z'FEFD'         XMIT CONTROL, FRAME 2, CHAR 3
                 *
0730   6D79              DC      =A'MY'          LABEL
0731   1300              DC      X'1300'         EXPECTED STATUS FLAG
0732   0086              DC      X'86'           CCP CONTROL FLAG
0733   0042              DC      66              RANGE, 1ST FRAME
0734   0003              DC      3               RANGE, 2ND FRAME
0735   FEFE              DC      Z'FEFE'         XMIT CONTROL, FRAME 2, CHAR 2
                 *
```

```
0736  6D7A                 DC     =A'MZ'             LABEL
0737  1300                 DC     X'1300'            EXPECTED STATUS FLAG
0738  0086                 DC     X'86'              CCP CONTROL FLAG
0739  0042                 DC     66                 RANGE, 1ST FRAME
073A  0004                 DC     4                  RANGE, 2ND FRAME
073B  FEFD                 DC     Z'FEFD'            XMIT CONTROL, FRAME 2, CHAR 3
                   *
073C  6D31                 DC     =A'M1'             LABEL
073D  2300                 DC     X'2300'            EXPECTED STATUS FLAG
073E  0084                 DC     X'84'              CCP CONTROL FLAG
073F  0043                 DC     67                 RANGE, 1ST FRAME
0740  0001                 DC     1                  RANGE, 2ND FRAME
0741  FEFF                 DC     Z'FEFF'            XMIT CONTROL, FRAME 2, CHAR 1
                   *
0742  6D32                 DC     =A'M2'             LABEL
0743  2500                 DC     X'2500'            EXPECTED STATUS FLAG
0744  0084                 DC     X'84'              CCP CONTROL FLAG
0745  0043                 DC     67                 RANGE, 1ST FRAME
0746  0004                 DC     4                  RANGE, 2ND FRAME
0747  FEFC                 DC     Z'FEFC'            XMIT CONTROL, FRAME 2, CHAR 4
                   *
0748  6D33                 DC     =A'M3'             LABEL
0749  2300                 DC     X'2300'            EXPECTED STATUS FLAG
074A  0086                 DC     X'86'              CCP CONTROL FLAG
074B  0043                 DC     67                 RANGE, 1ST FRAME
074C  0003                 DC     3                  RANGE, 2ND FRAME
074D  FEFE                 DC     Z'FEFE'            XMIT CONTROL, FRAME 2, CHAR 2
                   *
074E  6D34                 DC     =A'M4'             LABEL
074F  2500                 DC     X'2500'            EXPECTED STATUS FLAG
0750  0084                 DC     X'84'              CCP CONTROL FLAG
0751  0046                 DC     70                 RANGE, 1ST FRAME
0752  0004                 DC     4                  RANGE, 2ND FRAME
0753  FEFC                 DC     Z'FEFC'            XMIT CONTROL, FRAME 2, CHAR 4
                   *
0754  6D35                 DC     =A'M5'             LABEL
0755  2300                 DC     X'2300'            EXPECTED STATUS FLAG
0756  0086                 DC     X'86'              CCP CONTROL FLAG
0757  0046                 DC     70                 RANGE, 1ST FRAME
0758  0005                 DC     5                  RANGE, 2ND FRAME
0759  FEFC                 DC     Z'FEFC'            XMIT CONTROL, FRAME 2, CHAR 4
                   *
075A  6D36                 DC     =A'M6'
075B  7100                 DC     X'7100'            EXPECTED STATUS FLAG
075C  0086                 DC     X'86'              CCP FLAG
075D  0047                 DC     71                 RANGE 1
075E  0004                 DC     4                  RANGE 2
075F  FFBA                 DC     Z'FFBA'            XMIT CONTROL, FRAME 1, CHAR 68
                   *
0760  6D37                 DC     =A'M7'             LABEL
0761  1300                 DC     X'1300'            EXPECTED STATUS FLAG
0762  0086                 DC     X'86'              DELAYED TURN ON OF RECEIVE
0763  003F                 DC     63                 RANGE 1
0764  0005                 DC     5                  RANGE 2
0765  FEFC                 DC     Z'FEFC'            XMIT CONTROL, FRAME 2, CHAR 4
                   *
                   * TEST M8 TESTS RECEIVER RE-SYNC
                   *
0766  6D38                 DC     =A'M8'
0767  4130                 DC     Z'4130'            EXPECTED STATUS FLAG
0768  00E4                 DC     X'E4'
0769  0028                 DC     40                 RANGE1
076A  0004                 DC     4                  RANGE 2
076B  FE00                 DC     Z'FE00'             FRAME 2, CHAR = DON'T CARE
                   *
                   *
                   *
076C  0000                 RESV   18,0               ROOM FOR MORE
077E  0000         DEX     DC     0                  STORAGE FOR INDEX
                   *---------------------------------------------------------------
                   *
                   *   START OF 3 FRAME TESTS
                   *
                   *       THE FOLLOWING TESTS (EN -EP) ALL TRANSMIT 3 FRAMES
                   *       BEFORE RCV IS TURNED ON.
                   *
077F  9870 454E    RCV-EN LDR     $R1,=A'EN'
0781  9F00 11EA           STR     $R1,<ERMG+1        REPORT TEST
0783  9870 1150           LDR     $R1,=Z'1150'       EXPECTED STATUS
0785  9F00 117B           STR     $R1,<EXST
                   *
0787  B380 0D3D           LNJ     $B3,<FACDTA        FILL ASCENDING DATA
0789  8700 1174           CL      <HEAD              CLEAR HEADER FLAG
078B  AB80 0793           LAB     $B2,<PAR17
078D  AF80 0ECF           STB     $B2,<PARPTR
078F  C380 0D80           LNJ     $B4,<GHEAD         GENERATE HEADER
                   *
                   *
0791  A380 0A4A           LNJ     $B2,<RET           RECEIVE END TEST
0793  000C         PAR17  DC      =X'C'              RCV CONFIGURATION, 8 BITS
0794  000C                DC      =X'C'              XMIT CONFIG,  8 BITS
0795  008C                DC      X'8C'              FLAG IDLE BETWEEN FRAMES
0796  0045                DC      =X'45'             LR2 CONTROL, TEST, XMIT
0797  0084                DC      X'84'              DELAYED TURN ON OF RECEIVE
0798  0000                DC      0
0799  FD00                DC      Z'FD00'            RXMIT CONTROL AT FRAME 3
                   *
079A  1800                DC      <SDB               SEND BUFFER
079B  0040                DC      64                 RANGE IN BYTES
079C  0040                DC      X'40'              VALID
                   *
079D  1840                DC      <SDB+X'40'         BUFFER 2
079E  0002                DC      2                  RANGE
079F  0040                DC      X'40'              VALID
                   *
07A0  183C                DC      <SDB+60            BUFFER 3
07A1  0002                DC      2                  RANGE, FRAME 3
07A2  0060                DC      X'60'              LAST, VALID
                   *
07A3  8386                JMP     $B6
                   *---------------------------------------------------------------
07A4  9870 454F    RCV-EO LDR     $R1,=A'EO'
07A6  9F00 11EA           STR     $R1,<ERMG+1        REPORT TEST
```

```
07A8   9870 1250              LDR     $R1,=Z'1250'            EXPECTED STATUS
07AA   9F00 117B              STR     $R1,<EXST
                      *
07AC   B380 0D3D              LNJ     $B3,<FACDTA             FILL ASCENDING DATA
07AE   8700 1174              CL      <HEAD                   CLEAR HEADER FLAG
07B0   AB80 07B8              LAB     $B2,<PAR18
07B2   AFC0 071C              STB     $B2,PARPTR
07B4   C380 0D80              LNJ     $B4,<GHEAD              GENERATE HEADER
                      *
                      *
07B6   A380 0A4A              LNJ     $B2,<RET                RECEIVE END TEST
07B8   000C           PAR18   DC      =X'C'                   RCV CONFIGURATION, 8 BITS
07B9   000C                   DC      =X'C'                   XMIT CONFIG,  8 BITS
07BA   008C                   DC      X'8C'                   FILL WITH FLAGS
07BB   0045                   DC      =X'45'                  LR2 CONTROL, TEST, XMIT
07BC   0084                   DC      =X'84'                  DELAYED TURN ON OF RECEIVE
07BD   0000                   DC      0
07BE   FD00                   DC      Z'FD00'                 XMIT CONTROL, FRAME 1,
                      *
07BF   1800                   DC      <SDB                    SEND BUFFER
07C0   0040                   DC      64                      RANGE IN BYTES
07C1   0040                   DC      X'40'                   VALID
                      *
07C2   1840                   DC      <SDB+X'40'              BUFFER 2
07C3   0003                   DC      3                       RANGE
07C4   0040                   DC      X'40'                   VALID
                      *
07C5   1860                   DC      <SDB+X'60'              BUFFER 3
07C6   0002                   DC      2                       RANGE, FRAME 3
07C7   0060                   DC      X'60'                   LAST, VALID
                      *
07C8   8386                   JMP     $B6
                      *---------------------------------------------------------------------
07C9   9870 4550      RCV-EP  LDR     $R1,=A'EP'
07CB   9F00 11EA              STR     $R1,<ERMG+1             REPORT TEST
07CD   9870 2530              LDR     $R1,=Z'2530'            EXPECTED STATUS
07CF   9F00 117B              STR     $R1,<EXST
                      *
07D1   B380 0D3D              LNJ     $B3,<FACDTA             FILL ASCENDING DATA
07D3   8700 1174              CL      <HEAD                   CLEAR HEADER FLAG
07D5   AB80 07DD              LAB     $B2,<PAR19
07D7   AF80 0ECF              STB     $B2,<PARPTR
07D9   C380 0D80              LNJ     $B4,<GHEAD              GENERATE HEADER
                      *
                      *
07DB   A380 0A4A              LNJ     $B2,<RET                RECEIVE END TEST
07DD   000C           PAR19   DC      =X'C'                   RCV CONFIGURATION, 8 BITS
07DE   000C                   DC      =X'C'                   XMIT CONFIG,  8 BITS
07DF   008C                   DC      X'8C'                   FILL WITH FLAGS
07E0   0045                   DC      =X'45'                  LR2 CONTROL, TEST, XMIT
07E1   0084                   DC      X'84'                   DELAYED TURN ON OF RCV
07E2   0000                   DC      0
07E3   FD00                   DC      Z'FD00'                 XMIT CONTROL, FRAME  2
                      *
07E4   1800                   DC      <SDB                    SEND BUFFER
07E5   0043                   DC      67                      RANGE IN BYTES
07E6   0040                   DC      X'40'                   VALID
                      *
07E7   1840                   DC      <SDB+X'40'              BUFFER 2
07E8   0002                   DC      2                       RANGE
07E9   0040                   DC      X'40'                   VALID
                      *
07EA   1860                   DC      <SDB+X'60'              BUFFER 3
07EB   0002                   DC      2                       RANGE, FRAME 3
07EC   0060                   DC      X'60'                   LAST, VALID
                      *
07ED   8386                   JMP     $B6
                      *
                      *
                      *---------------------------------------------------------------------
                      * MEASURE TEST CLOCK AND PRINT. (1ST PASS ONLY)
                      *
07EE   9870 2057      SPED-W  LDR     $R1,=A' W'
07F0   9F00 11EA              STR     $R1,<ERMG+1             REPORT TEST
                      *
07F2   9800 1182              LDR     $R1,<PASSC
07F4   1900 07F7              BEZ     $R1,<SPED1
07F6   8386                   JMP     $B6                     NOT FIRST PASS
                      *
07F7   8980 1179      SPED1   CMZ     <LOOP
07F9   0902                   BE      >SPED2
07FA   8386                   JMP     $B6                     NOT FIRST LOOP
                      *
                      * FILL SEND BUFFER WITH ZERO'S
                      SPED2   CALL    ZV$F,SDB,C0,R400
07FB   FBC0 0003
07FD   D380 0000       X
07FF   0F80
0800   1800
0801   118E
0802   118B
                      *
0803   8700 116A              CL      <CHSZ                   SET FOR 8 BIT BYTES
0805   AB80 080B              LAB     $B2,<PAR21
0807   AF80 0ECF              STB     $B2,<PARPTR
0809   A380 09E7              LNJ     $B2,<SPTS               SPEED TEST
080B   000C           PAR21   DC      X'C'                    RCV CHAR = 8
080C   000C                   DC      X'C'                    XMIT CHAR = 8 BITS
080D   000C                   DC      X'C'                    FILL WITH ABORTS
080E   0047                   DC      =X'47'                  LR2 CONTROL
080F   0000                   DC      0                       FLAG
0810   0000                   DC      0                       RC ACTION CHR, FRAME
0811   0000                   DC      0                       XMIT ACTION CHR, FRAME
                      *
0812   1800                   DC      <SDB                    DATA ADDRESS
0813   0800                   DC      2048                    RANGE IN BYTES
0814   0060                   DC      X'60'                   VALID, LAST
                      *
0815   0F66                   NOP     >SPED2
0816   8386                   JMP     $B6                     DONE WITH TEST
                      *---------------------------------------------------------------------
                      *
                      * RANDOM DATA TEST
```

```
                                *
0817  9870 5244                 RAN-RD LDR    $R1,=A'RD'
0819  9F00 11EA                        STR    $R1,<ERMG+1        REPORT TEST
                                *
                                * GENERATE RANDOM DATA
                                *
081B  9800 1182                        LDR    $R1,<PASSC         GET PASS COUNT
081D  9B70 0040                        MUL    $R1,=64
081F  A800 1179                        LDR    $R2,<LOOP
0821  2F04                             MLV    $R2,=4
0822  9A52                             ADD    $R1,=$R2           GENERATE BASE FOR RAN GEN
0823  9F00 1189                        STR    $R1,<BASE
0825  9870 0100                        LDR    $R1,=X'100'
0827  9F00 0853                        STR    $R1,<RD4           SET UP SHORT RANGE
0829  9870 07FE                        LDR    $R1,=X'7FE'
082B  A800 1192                        LDR    $R2,<SPEED         GET SPEED OF LINE
082D  2906                             BEZ    $R2,>RD3           BRANCH IF NOT MEASURED YET
082E  A970 010E                        CMR    $R2,=270
0830  0383                             BLE    >RD3
0831  9F00 0853                        STR    $R1,<RD4           SET UP LONG RANGE
                                *
                                RD3    CALL   ZV$FI,BASE,MODE    INITIALIZE
0833  FBC0 0003
0835  D380 0000        X
0837  0F80
0838  1189
0839  118A
                                *
                                       CALL   ZV$FR,SDB,R85      FILL X'85' PSUEDO RAN
083A  FBC0 0003
083C  D380 0000        X
083E  0F80
083F  1800
0840  118C
0841  AB80 084B                        LAB    $B2,<PAR20
0843  AF80 0ECF                        STB    $B2,<PARPTR
0845  8700 1174                        CL     <HEAD              CLEAR HEADER FLAG
0847  C380 0D80                        LNJ    $B4,<GHEAD         GENERATE HEADER
                                *
                                *
0849  A380 08CD                        LNJ    $B2,<LPTS          LOOP TEST
084B  000C                      PAR20  DC     =X'C'              RCV CONFIG, 8 BITS
084C  000C                             DC     =X'C'              XMIT CONFIG
084D  000C                             DC     X'C'               FILL WITH ABORTS
084E  0047                             DC     =X'47'             LR2 - RCV, XMIT, TEST
084F  0000                             DC     0                  ACTION FLAG
0850  0000                             DC     0                  RCV ACTION FRAME, CHAR
0851  0000                             DC     0                  XMIT ACTION FRAME, CHAR
                                *
0852  1800                             DC     <SDB               DATA
0853  0100                      RD4    DC     X'100'             RANGE
0854  0060                             DC     X'60'              LAST,V
                                *
0855  8386                             JMP    $B6
                                *-----------------------------------------------------------------
                                *
                                * PARTIAL BYTE, CRC TEST, BIT INVERSION TEST
                                *
                                *   THE FOLLOWING TEST REPORTS ERRORS WITH LABEL "XZ" WHERE
                                *   Z IS ANY CHAR.  LOOK FOR ENTRY "XZ" IN TABLE "CRCTST" FOR
                                *   SPECIFICS OF A FAILING TEST.
                                *
                                *
0856  1CEC                      PCRC-X LDV    $R1,=-20
0857  9F00 1169                 PCRREP STR    $R1,<COUNT         DO TEST 20 TIMES
0859  8751                             CL     =$R1               CLEAR INDEX TO PARAMETER TABLE
                                *
085A  9B80 08AF                 CRC-LP LAB    $B1,<CRCTST
085C  2C01                             LDV    $R2,=1
                                *
085D  C2DD                             LLH    $R4,$B1.+$R1       GET XMIT LR7
085E  CF00 089D                        STR    $R4,<PCRC1
0860  C2DD                             LLH    $R4,$B1.+$R1       GET RCV LR6
0861  CF00 089C                        STR    $R4,<PCRC2
0863  C2DD                             LLH    $R4,$B1.+$R1       GET FLAG
0864  CF00 08A0                        STR    $R4,<PCRC3
0866  C2DD                             LLH    $R4,$B1.+$R1       GET PARTIAL BYTE INFO
0867  CF00 089E                        STR    $R4,<PCRC4
0869  C2DD                             LLH    $R4,$B1.+$R1       GET ERROR LABLE BYTE 1
086A  C780 11EA                        STH    $R4,<ERMG+1
086C  4900 08AA                        BEZ    $R4,<ENDCRC        BR IF END OF LIST
086E  C2DD                             LLH    $R4,$B1.+$R1       GET ERROR LABLE BYTE 2
086F  C7A0 11EA                        STH    $R4,<ERMG+1.$R2
0871  8700 1175                        CL     <IFLG              CLEAR BIT INVERSION FLAG (XMIT)
                                *
0873  8280 089D                        LB     <PCRC1,=X'20'      GET TCB BIT
0875  0020
0876  0580 0884                        BBF    <PCRC5             BRANCH IF NOT SET
                                *
0878  9870 0304                        LDR    $R1,=Z'0304'       LCF 4 LONG, TCB = 3
                                * IN THIS TEST THE ASSUMPTION IS MADE IF A TCB BIT IS SET,
                                * THEN THAT HALF OF THE LINE WILL HAVE BIT INVERSION.
087A  9F00 1174                        STR    $R1,<HEAD
087C  8A80 1175                        INC    <IFLG
                                *
087E  8280 089C                        LB     <PCRC2,=X'20'      GET TCB BIT (RCV)
0880  0020
0881  0583                             BBF    >PCRC5             BRANCH IF NOT SET
0882  8A80 1175                        INC    <IFLG              SET BIT INVERSION FLAG TO 2
                                *
                                * GENERATE RANDOM DATA FOR CRC TEST
                                *
                                PCRC5  CALL   ZV$FR,SDB,R10      FILL X'10' WORDS OF RANDOM DATA
0884  FBC0 0003
0886  D380 0000        X
0888  0F80
0889  1800
088A  118D
088B  9F00 077E                        STR    $R1,<DEX           STORE INDEX
088D  AB80 089C                        LAB    $B2,<PCRC2
088F  AF80 0ECF                        STB    $B2,<PARPTR
0891  C380 0D80                        LNJ    $B4,<GHEAD         GENERATE HEADER
                                *
                                * GENERATE CRC'S AND CRC RESIDUES AND TACK ON END OF SEND DATA
                                *
0893    C380 0D4D                       LNJ     $B4,<GCRC

0895    8980 1175               PCRC7   CMZ     <IFLG                   CHECK BIT INVERSION FLAG
0897    0903                            BE      >PCRC8                  BRANCH IF NOT SET
0898    C380 0E0A                       LNJ     $B4,<IVRT               INVERT DATA
089A    A380 08CD               PCRC8   LNJ     $B2,<LPTS               LOOP TEST
089C    0000                    PCRC2   DC      0                       RCV LR6
089D    0000                    PCRC1   DC      0                       XMIT LR7
089E    0000                    PCRC4   DC      0                       FILL WITH ABORTS
089F    0047                            DC      =X'47'                  LR2 CONTROL, TEST, XMIT
08A0    0000                    PCRC3   DC      0                       FLAG
08A1    0000                            DC      0                       XMIT ACTION FLAG, FRAME
08A2    0000                            DC      0                       RCV ACTION FLAG, FRAME
                                *
08A3    1800                            DC      <SDB
08A4    001F                            DC      X'1F'                   RANGE
08A5    0060                            DC      X'60'                   VALID
                                *
08A6    9800 077E                       LDR     $R1,<DEX
08A8    0F80 085A                       B       <CRC-LP
                                *
08AA    9800 1169               ENDCRC  LDR     $R1,<COUNT              GET TEST COUNT
08AC    1780 0857                       BINC    $R1,<PCRREP             DO AGAIN IF NOT  DONE
08AE    8386                            JMP     $B6
                                *
                                * TABLE FOR  CRC, PARTIAL BYTE SIZE TEST
                                *
                                *   WORD 1      BITS 0 - 7      LR7, XMIT
                                                BITS 8 - 15     LR6, RCV
                                *
                                *   WORD 2      BITS 0 - 7      CCP FLAG
                                *               BITS 8 - 15     XMIT BS 0,1,2 FOR LAST BYTE
                                *
                                * TEST XA
                                *   WORD 3      BITS 0 - 15     ASCII TEST IDENTIFIER
                                *
                                *
                                *  TEST XA READS IN CRC RESIDUES BY SPECIAL RCV BYTE CODE OF "001"
                                *
08AF    0C0C                    CRCTST  DC      Z'0C0C'                 XMIT, RCV = 8 BITS
08B0    080C                            DC      Z'080C'                 TEST, 8 BIT LAST
08B1    5841                            DC      Z'5841'                 XA
                                *
                                * TEST XB
                                *
08B2    0C0C                            DC      Z'0C0C'                 XMIT, RCV = 8 BITS
08B3    000F                            DC      Z'000F'                 4 BIT LAST BYTE
08B4    5842                            DC      Z'5842'                 XB
                                * TEST XC
                                *
08B5    0808                            DC      Z'0808'                 XMIT, RCV = 7 BITS
08B6    000B                            DC      Z'000B'                 3 BIT RESIDUE
08B7    5843                            DC      Z'5843'                 XC
                                * TEST XG
                                *
08B8    0404                            DC      Z'0404'                 XMIT, RCV = 6 BITS
08B9    0007                            DC      Z'0007'                 2 BIT RESIDUE
08BA    5847                            DC      Z'5847'                 XG
                                * TEST XH
                                *
08BB    0000                            DC      Z'0000'                 XMIT, RCV = 5 BITS
08BC    0003                            DC      Z'0003'                 3 BIT RESIDUE
08BD    5848                            DC      Z'5848'                 XH
                                * THE FOLLOWING HAVE XMIT BIT INVERSION
                                *
                                * TEST XI
08BE    2C00                            DC      Z'2C00'                 8 BITS, TCB FOR XMIT
08BF    080D                            DC      Z'080D'                 TEST 8 BIT LAST
08C0    0000                            DC      Z'0'                    XI TEMP!!! SB 5849
                                * TEST XJ
08C1    2C00                            DC      Z'2C00'                 8  BITS, TCB FOR TRANSMIT
08C2    080B                            DC      Z'080B'                 TEST, 3 BIT RESIDUE
08C3    584A                            DC      Z'584A'                 XJ
                                *
                                * TEST XK
08C4    2C20                            DC      Z'2C20'                 8 BITS, TCB ON XMIT, RCV
08C5    000C                            DC      Z'000C'                 NO RESIDUE BYTE
08C6    584B                            DC      Z'584B'
                                * TEST XL
08C7    2C20                            DC      Z'2C20'                 8 BITS, TCB ON XMIT, RCV
08C8    0001                            DC      Z'0001'                 5 BYTE RESIDUE
08C9    584C                            DC      Z'584C'
                                *
08CA    0000                            RESV    3,0                     END
                                *
                                *
                                *
                                *-----------------------------------------------------------------
                                *
                                * BASIC DATA LOOP TEST
                                *
08CD    B380 0E32               LPTS    LNJ     $B3,<PASS               PASS PARAMETERS
08CF    8F00 2063                       SAVE    <SAVMAJ,=Z'0642'
08D1    0642
08D2    8753                            CL      =$R3                    CLEAR CHANNEL
08D3    C380 0ED0               LCT2    LNJ     $B4,<FLN                FIND ACTIVE LINE
08D5    0F82                            B       >LPTS2                  NO MORE ACTIVE LINES
08D6    0F85                            B       >LCT8
08D7    8F80 2063               LPTS2   RSTR    <SAVMAJ,=Z'0642'
08D9    0642
08DA    8382                            JMP     $B2                     EXIT TEST
                                *
08DB    BF00 1166               LCT8    STR     $R3,<CHAN               STORE CHAN TO TEST
08DD    8780 0BFB                       CLH     <XTEMP                  SET TO NOT READ XMIT FW REV
08DF    B380 0BA3                       LNJ     $B3,<LOAD               LOAD MLCP AND EXECUTE CCP
                                *
                                * READ RCV STATUS TO $R5, XFER SHOULD BE OVER
                                *
08E1    9800 0C1B                       LDR     $R1,<XRNG1              GET RANGE
08E3    8756                            CL      =$R6
08E4    9970 0002                       CMR     $R1,=2
```

```
08E6  0203                        BL    >LPTS1               BRANCH IF RANGE <2
08E7  E870 1010                   LDR   $R6,=X'1010'
08E9  D956                LPTS1   CMR   $R5,=$R6
08EA  0903                        BE    >$+3
08EB  E3C0 06C2                   LNJ   $B6,ERRDB            RCV STATUS WRONG
08ED  C380 1067                   LNJ   $B4,<DLAYLG          DELAY FOR ILS RUPT
                          *
                          * READ P VALUE FOR RCV
                          *
08EF  C380 0F6B                   LNJ   $B4,<RPVLU           READ P TO R5
08F1  E870 0203                   LDR   $R6,=X'203'          SB
08F3  9800 11EA                   LDR   $R1,<ERMG+1          GET TEST
08F5  9970 204F                   CMR   $R1,=A' O'
08F7  0905                        BE    >LPTS3               BRANCH IF TEST 0
08F8  D956                        CMR   $R5,=$R6
08F9  0903                        BE    >LPTS3               BRANCH MEANS P IS GOOD
08FA  E3C0 06B3                   LNJ   $B6,ERRDB            RCV P COUNTER BAD AFTER FRAME RCV
08FC  8700 1176           LPTS3   CL    <MASK
08FE  9C80 0C1A                   LDB   $B1,<XADD1           GET ADDRESS OF DATA
0900  9F80 092E                   STB   $B1,<COMP+4+$AF
                          *
                          * READ TRANSMIT STATUS TO R5
                          *
0902  8AD3                        INC   =$R3                 GET XMIT CHANNEL
0903  C3C0 075C                   LNJ   $B4,DLAY             DELAY 25 MS
                          *
0905  9800 0C28                   LDR   $R1,<XADD2           GET ADDRESS OF SECOND CCB
0907  1900 0911                   BEZ   $R1,<RSTLST          IF ZERO READ LAST STATUS
0909  E870 1000                   LDR   $R6,=X'1000'         SHOULD BE STATUS
090B  C380 10D9                   LNJ   $B4,<INXT            INPUT NEXT STATUS
090D  D956                        CMR   $R5,=$R6
090E  0903                        BE    >$+3
090F  E3C0 069E                   LNJ   $B6,ERRDB            STATUS WRONG, FIRST CCB
                          *
0911  C380 10D9           RSTLST  LNJ   $B4,<INXT            GET NEXT STATUS
0913  E870 1020                   LDR   $R6,=X'1020'         SHOULD BE STATUS
0915  D956                        CMR   $R5,=$R6             COMPARE WITH IS
0916  0903                        BE    >$+3
0917  E3C0 0696                   LNJ   $B6,ERRDB            STATUS ERROR, TRANSMIT
                          *
                          * READ P VALUE FOR XMIT
                          *
0919  C380 0F6B                   LNJ   $B4,<RPVLU           READ P TO R 5
091B  E870 0403                   LDR   $R6,=X'403'          GET SB
091D  E955                        CMR   $R6,=$R5
091E  090B                        BE    >COMP                BRANCH IF GOOD
091F  E3C0 068E                   LNJ   $B6,ERRDB            PVALUE WRONG FOR CCP, XMIT
0921  9800 1175                   LDR   $R1,<IFLG            GET BIT INVERSION FLAG
0923  8700 1175                   CL    <IFLG                CLEAR FLAG
0925  1D01                        CMV   $R1,=1
0926  0983                        BNE   >COMP                BRANCH IF NOT SET
0927  C380 0E0A                   LNJ   $B4,<IVRT            INVERT DATA
                          *
                          *
                          * COMPARE DATA
                          *
                          COMP    CALL    ZV$C,$,RTB,MASK,RANGE,ERAR
0929  FBC0 0003
092B  D380 0000        X
092D  0F80
092E  0929
092F  1C00
0930  1176
0931  116C
0932  11BD
0933  8980 11BD                   CMZ   <ERAR
0935  0909                        BE    >TST1C
0936  D800 11BF                   LDR   $R5,<ERAR+2          IS
0938  E800 11BE                   LDR   $R6,<ERAR+1          SB
093A  F800 11BD                   LDR   $R7,<ERAR            WORD NUMBER
093C  E3C0 0671                   LNJ   $B6,ERRDB            DATA ERROR IN XFER
093E  8280 119B           TST1C   LB    <ODDFLG,=1
0940  0001
0941  0581 0019                   BBF   TST1D                BRANCH IF EVEN RANGE
0943  9800 116C                   LDR   $R1,<RANGE
0945  1001                        SOL   $R1,1                FORM RANGE IN BYTES
0946  A080 0BEF                   LDH   $R2,<PARBYT          GET BYTE SIZE OF LAST
0948  2041                        SOR   $R2,1
0949  C820 04AD                   LDR   $R4,<TCBBSZ.$R2      CONVERT BYTE CODE TO BINARY
094B  A854                        LDR   $R2,=$R4
094C  C820 0E02                   LDR   $R4,<DMASK.$R2       GET  MASK
094E  CF40 0827                   STR   $R4,MASK
0950  E290 1800                   LLH   $R6,<SDB.$R1         GET WHAT DATA SHOULD BE
0952  E500 1176                   AND   $R6,<MASK
0954  D290 1C00                   LLH   $R5,<RTB.$R1         GET WHA DATA IS
0956  D956                        CMR   $R5,=$R6
0957  0901 0003                   BE    TST1D
0959  E3C0 0654                   LNJ   $B6,ERRDB            LAST DATA BYTE INCORRECT
                          *
095B  8280 0BF7           TST1D   LB    <ACTFLG,=Z'0800'     GET TEST BIT FLAG
095D  0800
095E  058D                        BBF   >TST1E               BRANCH IF NO CRC TO CHECK
                          *
                          * CHECK CRC'S AND RESIDUES
                          *
095F  8AD1                        INC   =$R1                 BUMP TO NEXT BYTE
0960  2CFC                        LDV   $R2,=-4
0961  E290 1800           TST1G   LLH   $R6,<SDB.$R1
0963  D290 1C00                   LLH   $R5,<RTB.$R1
0965  8AD1                        INC   =$R1
0966  D956                        CMR   $R5,=$R6             COMPARE IS VS SB
0967  0903                        BE    >TST1F
0968  E3C0 0645                   LNJ   $B6,ERRDB            ERROR IN CRC OR CRC RESIDUE
096A  27F7                TST1F   BINC  $R2,>TST1G
                          *
096B  B800 1166           TST1E   LDR   $R3,<CHAN            GET TESTED CHANNEL
096D  3E04                        ADV   $R3,=4               BUMP CHANNEL NUMBER
096E  0F80 08D3                   B     <LCT2
                          *----------------------------------------------------------------
                          *
                          * UNDERUN, ABORT TEST
                          *
0970  B380 0E32           ABUND   LNJ   $B3,<PASS            PASS PARAMETERS
```

```
0972  8F00 2063           SAVE   <SAVMAJ,=Z'0042'
0974  0042
0975  8753                CL     =$R3
0976  C380 0ED0   ABUND2  LNJ    $B4,<FLN               FIND ACTIVE LINE
0978  0F82                B      >ABUND6                NO MORE LINES
0979  0F85                B      >ABUND3
097A  8F80 2063   ABUND6  RSTR   <SAVMAJ,=Z'0042'
097C  0042
097D  8382                JMP    $B2
                  *
097E  B380 0BA3   ABUND3  LNJ    $B3,<LOAD              LOAD AND EXECUTE CHANNEL PROGRAM
                  *
                  *  XFER SHOULD BE OVER, ABORT SHOULD BE SET (IF DATA + FCS > 24 BITS)
                  *
0980  8756                CL     =$R6
0981  7C03                LDV    $R7,=3
0982  9280 11EA           LLH    $R1,<ERMG+1            GET TEST MESSAGE
0984  1D4A                CMV    $R1,=X'4A'
0985  0982                BNE    >ABUND8
0986  7C05                LDV    $R7,=5
0987  8280 0BF7   ABUND8  LB     <ACTFLG,=Z'1000'       OVERUN
0989  1000
098A  0583                BBF    >ABUND7                BRANCH IF NO OVERUN
098B  E870 2000           LDR    $R6,=Z'2000'           SET OVERUN BIT
098D  9800 0C1B   ABUND7  LDR    $R1,<XRNG1
098F  9957                CMR    $R1,=$R7
0990  0380 0994           BLE    <ABUND4                BRANCH IF RANGE <3
0992  E470 1230           OR     $R6,=X'1230'           SHOULD BE STATUS
0994  D956        ABUND4  CMR    $R5,=$R6               COMPARE WITH IS
0995  0903                BE     >$+3
0996  E3C0 0617           LNJ    $B6,ERRDB              RCV STATUS WRONG
                  *
                  * READ XMIT STATUS TO R5
                  *
0998  8AD3                INC    =$R3                   BUMP CHANNEL NUMBER
0999  C380 1067           LNJ    $B4,<DLAYLG            DELAY 225 MS
099B  C380 10D9           LNJ    $B4,<INXT              INPUT NEXT STATUS
                  *
                  * STATUS TEST FOR  RECEIVE
                  *
099D  8280 0BF7           LB     <ACTFLG,=Z'0500'       CHECK UNDERUN EXPECTED
099F  0500
09A0  0584                BBF    >ABUND1
09A1  E870 3020           LDR    $R6,=Z'3020'           SB STATUS
09A3  0F83                B      >ABUND5
09A4  E870 1020   ABUND1  LDR    $R6,=Z'1020'           COMPLETE, LAST
09A6  D956        ABUND5  CMR    $R5,=$R6
09A7  0903                BE     >$+3
09A8  E3C0 0605           LNJ    $B6,ERRDB              XMIT STATUS BAD, SINGLE FRAME
09AA  CB88 0ECF           LAB    $B4,*<PARPTR           GET POINTER TO PARAMETERS
09AC  9844 0006           LDR    $R1,$B4.6              PICK UP XMIT ABORT COUNT
09AE  9470 FF00           OR     $R1,=Z'FF00'
09B0  8251                NEG    =$R1                   MAKE POSITIVE
09B1  1EFE                ADV    $R1,=-2
09B2  9F00 0C1B           STR    $R1,<XRNG1             VALID DATA RANGE
09B4  7C03                LDV    $R7,=3
09B5  D280 11EA           LLH    $R5,<ERMG+1            GET TEST LABLE
09B7  5D4A                CMV    $R5,=X'4A'
09B8  0982                BNE    >ABUND9                BRANCH IF NOT TEST J
09B9  7C05                LDV    $R7,=5
09BA  9957        ABUND9  CMR    $R1,=$R7
09BB  0AA7                BALE   >UND2                  DON'T COMPARE DATA FOR ILLEGAL CASE
09BC  1001                SOL    $R1,1                  FORM RANGE IN WORDS
09BD  9F00 116C           STR    $R1,<RANGE
                  *
09BF  8700 1176           CL     <MASK
                  COMP1   CALL   ZV$C,$,$,MASK,RANGE,ERAR
09C1  FBC0 0003
09C3  D380 0000     X
09C5  0F80
09C6  09C1
09C7  09C1
09C8  1176
09C9  116C
09CA  11BD
09CB  8980 11BD           CMZ    <ERAR
09CD  0909                BE     >UND3
09CE  D800 11BF           LDR    $R5,<ERAR+2            IS
09D0  E800 11BE           LDR    $R6,<ERAR+1            SB
09D2  E800 11BD           LDR    $R6,<ERAR              WORD NUMBER
09D4  E3C0 05D9           LNJ    $B6,ERRDB              DATA ERROR
09D6  9800 0C1B   UND3    LDR    $R1,<XRNG1
09D8  88D1                DEC    =$R1
09D9  8755                CL     =$R5
09DA  D956                CMR    $R5,=$R6
09DB  0907                BE     >UND2
09DC  E3C0 05D1           LNJ    $B6,ERRDB              LAST BYTE TRANSFERED WRONG
09DE  0F80 09E2           B      <UND2                  DONE
09E0  D090 1C00           LDH    $R5,<RTB.$R1           GET ACTUAL LAST BYTE
09E2  B800 1166   UND2    LDR    $R3,<CHAN              GET TESTED CHANNEL
09E4  3E04                ADV    $R3,=4                 BUMP CHANNEL NUMBER
09E5  0F80 0976           B      <ABUND2
                  *
                  *------------------------------------------------------------
                  *
                  * SPEED TEST
                  *
09E7  B380 0E32   SPTS    LNJ    $B3,<PASS              PASS PARAMETERS
09E9  8F00 2063           SAVE   <SAVMAJ,=Z'0002'
09EB  0002
09EC  8753                CL     =$R3
09ED  C380 0ED0   SPTS1   LNJ    $B4,<FLN               FIND LINE NUMBER
09EF  0F83                B      >SPTS3                 NO MORE LINES
                  *
09F0  0F80 09F6           B      <SPTS2
09F2  8F80 2063   SPTS3   RSTR   <SAVMAJ,=Z'0002'
09F4  0002
09F5  8382                JMP    $B2                    EXIT
                  *
09F6  BF00 1166   SPTS2   STR    $R3,<CHAN
09F8  B380 0BA3           LNJ    $B3,<LOAD              LOAD MLCP + EXECUTE CCP
09FA  9870 0804           LDR    $R1,=2052              RANGE + 2 FLAGS +ILS
09FC  9200 1183           SUB    $R1,<RRNG              GETS NUMBER ACTUALLY SENT
```

```
09FE  1E03                     ADV    $R1,=3              COMPENSATE FOR BUFFERING
09FF  1880 0A02                BGEZ   $R1,<SPED3
0A01  8751                     CL     =$R1                MAKE SURE +
0A02  9F00 1168       SPED3    STR    $R1,<CHRCNT
0A04  C870 01E0                LDR    $R4,=480            4000 MSEC (TICKS)
0A06  C200 0000    X           SUB    $R4,<ZHRTCC         REAL TIME CLOCK TICKS
0A08  CF00 1167                STR    $R4,<ELPS           ELAPSED TIME
0A0A  D380 111E                LNJ    $B5,<TKSEC          CONVERT RTC TICKS TO MSEC
0A0C  D380 1138                LNJ    $B5,<BPS            CALCULATE BITS/SEC
                      * PRINT LINE SPEED
0A0E  3041                     SOR    $R3,1               ALIGN LINE SPEED
0A0F  BF00 1177                STR    $R3,<TPR
                               CALL   ZV$TC,LINE          LINE
0A11  FBC0 0003
0A13  D380 0000    X
0A15  0F80
0A16  0A41
                               CALL   ZV$TD,TPR           NUMBER
0A17  FBC0 0003
0A19  D380 0000    X
0A1B  0F80
0A1C  1177
                               CALL   ZV$TC,SPDMSG        BITS/SEC =
0A1D  FBC0 0003
0A1F  D380 0000    X
0A21  0F80
0A22  0A44
                               CALL   ZV$TD,MSBS          BITS/SEC
0A23  FBC0 0003
0A25  D380 0000    X
0A27  0F80
0A28  1187
0A29  1C5C                     LDV    $R1,=X'5C'          NULL
0A2A  9F00 1177                STR    $R1,<TPR
0A2C  8700 1178                CL     <TEMP
                               CALL   ZV$TD,TEMP,TPR      ADD ZERO ON END
0A2E  FBC0 0003
0A30  D380 0000    X
0A32  0F80
0A33  1178
0A34  1177
                      *
0A35  9800 1187                LDR    $R1,<MSBS           GET SPEED
0A37  3042                     SOR    $R3,2               ALIGN FOR DAUGHT BD NUMB
0A38  9F30 11A0                STR    $R1,<HCLS-1.$R3     STORE AWAY SPEED/10
0A3A  9F00 1192                STR    $R1,<SPEED
                      *
0A3C  B800 1166                LDR    $R3,<CHAN           GET BACK TESTED CHANNEL
0A3E  3E04                     ADV    $R3,=4              BUMP CHANNEL NUMBER
0A3F  0F80 09ED                B      <SPTS1              DO NEXT CHANNEL
                      *
0A41  4C49 4E45 2400  LINE     TEXT   'LINE$'
0A44  4249 5453 2F53  SPDMSG   TEXT   'BITS/SEC = $'
0A47  4543 203D 2024
                      *-----------------------------------------------------------------
                      *
                      * RECEIVE END TEST
                      *
0A4A  0F00 0A52       RET      NOP    <RET1
0A4C  8F00 2063                SAVE   <SAVMAJ,=Z'0002'
0A4E  0002
0A4F  B380 0E32                LNJ    $B3,<PASS           PASS PARAMETERS
0A51  8753                     CL     =$R3
0A52  C380 0ED0       RET1     LNJ    $B4,<FLN            FIND ACTIVE LINE
0A54  0F83                     B      >RET9
0A55  0F80 0A5B                B      <RET2
0A57  8F80 2063       RET9     RSTR   <SAVMAJ,=Z'0002'
0A59  0002
0A5A  8382                     JMP    $B2
                      *
0A5B  BF00 1166       RET2     STR    $R3,<CHAN           STORE CHANNEL
                               CALL   ZV$F,STBL,CM1,C4    INITIALIZE STATUS TABLE
0A5D  FBC0 0003
0A5F  D380 0000    X
0A61  0F80
0A62  0AEE
0A63  1197
0A64  1196
                      *
0A65  B380 0BA3                LNJ    $B3,<LOAD           LOAD CHANNEL PROGRAM AND EXECUTE
                      *
                      * LOOP TO CHECK STATUS AND DATA FOR FRAMES SENT
                      *
0A67  1C01                     LDV    $R1,=1
0A68  9F00 1177                STR    $R1,<TPR            INITIALIZE LAST STATE
0A6A  8752                     CL     =$R2                TRACKS CCB NUMBER
                      *
0A6B  F800 117B                LDR    $R7,<EXST           GET EXP STAT WORD
0A6D  8756            FNULL    CL     =$R6
0A6E  7084                     DOL    $R7,4               SHIFT LEFT DIGIT TO R6
0A6F  6D04                     CMV    $R6,=4              CHECK FOR NULL ON FIRST
0A70  0985                     BNE    >RET14
0A71  8AD2                     INC    =$R2
0A72  EF00 1177                STR    $R6,<TPR            SET LAST STATE
0A74  0FF9                     B      >FNULL
                      *
0A75  DF20 0AEE       RET14    STR    $R5,<STBL.$R2       STORE STATUS FOR REFERENCE
0A77  D570 3250                AND    $R5,=Z'3250'        STRIP TO STATUS COMPLETE, OVRUN, EOF
0A79  6D01                     CMV    $R6,=1
0A7A  0922                     BE     >RET5               BRANCH IF GOOD DATA, STATUS
0A7B  EF00 1177       RETX     STR    $R6,<TPR            STORE LAST STATE
0A7D  EF54                     STR    $R6,=$R4
0A7E  8AD2                     INC    =$R2                BUMB CCB NUMBER
0A7F  4D04                     CMV    $R4,=4              CHECK FOR NULL
0A80  0900 0AD1                BE     <RET8               DO NEXT
0A82  4D03                     CMV    $R4,=3
0A83  0983                     BNE    >RET3               BRANCH IF NOT ZERO STATUS
0A84  8756                     CL     =$R6                EXPECT ZERO STATUS
0A85  0F90                     B      >RET11
0A86  4D02            RET3     CMV    $R4,=2
0A87  090C                     BE     >RET13              B = OVERRUN
0A88  E870 3000                LDR    $R6,=Z'3000'
0A8A  4D05                     CMV    $R4,=5
```

```
0A8B  090A                                BE     >RET11               BRANCH = MISSING FRAME
0A8C  E870 1210                           LDR    $R6,=Z'1210'
0A8E  4D06                                CMV    $R4,=6
0A8F  0906                                BE     >RET11               BRANCH IF ABORTED FRAME
0A90  E870 3210                           LDR    $R6,=Z'3210'
0A92  0F83                                B      >RET11               IS ABORTED, OVERUN FRAME
                                   *
0A93  E870 3010                    RET13  LDR    $R6,=Z'3010'
0A95  D956                         RET11  CMR    $R5,=$R6
0A96  0900 0AD1                           BE     <RET8                GOOD
0A98  E3C0 0515                    RET4   LNJ    $B6,ERRDB            STATUS ERROR - SHOULD BE IN R5
0A9A  0F80 0AD1                           B      <RET8
0A9C  E870 1010                    RET5   LDR    $R6,=Z'1010'         COMPLETE, EOF
0A9E  D956                                CMR    $R5,=$R6
0A9F  0903                                BE     >RET6                GOOD IF BRANCH
0AA0  E3C0 050D                           LNJ    $B6,ERRDB            ERROR, STATUS, RCV
                                   *
                                   * CHECK DATA
                                   *
0AA2  CCA0 0AEA                    RET6   LDB    $B4,<RADTBL.$R2      GET ADDRESS OF OUTPUT ADDRESS
0AA4  BCF4                                LDB    $B3,+$B4             GET ADDRESS
0AA5  BF80 0AC2                           STB    $B3,<CDATA+4+$AF
0AA7  E804                                LDR    $R6,$B4              GET RANGE
0AA8  6041                                SOR    $R6,1                GET RANGE IN WORDS
0AA9  EF00 116C                           STR    $R6,<RANGE
0AAB  9852                                LDR    $R1,=$R2
0AAC  E800 1177                           LDR    $R6,<TPR             GET LAST STATE
0AAE  6D04                                CMV    $R6,=4               SEE IF LAST STATE WAS NULL
0AAF  0982                                BNE    >RET10
0AB0  88D1                                DEC    =$R1                 SET TO GET LAST POINTER
0AB1  BC90 0AEA                    RET10  LDB    $B3,<RADTBL.$R1      GET RCV ADD -X'400'
0AB3  8AD2                                INC    =$R2                 BUMP FOR NEXT TIME
0AB4  CC83                                LDB    $B4,$B3              GET ADDRESS
0AB5  BBC4 0400                           LAB    $B3,$B4.X'400'
0AB7  BF80 0AC3                           STB    $B3,<CDATA+4+2*$AF
0AB9  8700 1176                           CL     <MASK
0ABB  CF00 1177                           STR    $R4,<TPR             SAVE LAST STATE
                                   *
                                   CDATA  CALL   ZV$C,$,$,MASK,RANGE,ERAR
0ABD  FBC0 0003
0ABF  D380 0000        X
0AC1  0F80
0AC2  0ABD
0AC3  0ABD
0AC4  1176
0AC5  116C
0AC6  11BD
0AC7  8980 11BD                           CMZ    <ERAR
0AC9  0908                                BE     >RET8
0ACA  D800 11BF                           LDR    $R5,<ERAR+2          IS
0ACC  E800 11BE                           LDR    $R6,<ERAR+1          SB
0ACE  E3C0 04DF                           LNJ    $B6,ERRDB            DATA ERROR IN XFER
0AD0  0000                                HLT
0AD1  9800 1192                    RET8   LDR    $R1,<SPEED
0AD3  9970 010E                           CMR    $R1,=270             COMPARE FOR 2700 BAUD
0AD5  0303                                BG     >RET15
0AD6  C380 1067                           LNJ    $B4,<DLAYLG          LONG DELAY
0AD8  C380 1067                    RET15  LNJ    $B4,<DLAYLG          DELAY 225MS
0ADA  8756                                CL     =$R6
0ADB  7084                                DOL    $R7,4                GET NEXT STATUS STATE
0ADC  89D6                                CMZ    =$R6                 CHECK IF DONE
0ADD  0908                                BE     >RET7                BRANCH IF DONE
0ADE  7D04                                CMV    $R7,=4               TEST FOR NULL
0ADF  0900 0A75                           BE     <RET14
0AE1  C380 10D9                           LNJ    $B4,<INXT            INPUT NEXT STATUS
0AE3  0F80 0A75                           B      <RET14
0AE5  B800 1166                    RET7   LDR    $R3,<CHAN            GET TESTED CHANNEL
0AE7  3E04                                ADV    $R3,=4               BUMP CHANNEL NUMBER
0AE8  0F80 0A52                           B      <RET1
                                   *
                                   * POINTERS TO XMIT ADDRESS AND RANGE
                                   *
0AEA  0C1A                         RADTBL DC     <XADD1
0AEB  0C28                                DC     <XADD2
0AEC  0C33                                DC     <XADD3
0AED  0C3E                                DC     <XADD4
                                   *
                                   * STATUS READ BACK STORED HERE
                                   *
0AEE  FFFF                         STBL   DC     -1
0AEF  FFFF                                DC     -1
0AF0  FFFF                                DC     -1
0AF1  FFFF                                DC     -1
                                   *---------------------------------------------------------------------
                                   *
                                   * INITIAL CONDITIONS TEST
                                   *
0AF2  B380 0E32                    TONTST LNJ    $B3,<PASS
0AF4  8F00 2063                           SAVE   <SAVMAJ,=Z'0002'
0AF6  0002
0AF7  8753                                CL     =$R3                 R3 TRACKS CHANNEL
0AF8  C380 0ED0                    TON1   LNJ    $B4,<FLN             FIND ACTIVE LINE
0AFA  0F82                                B      >TON2                NO MORE LINES
0AFB  0F85                                B      >TON3
                                   *
0AFC  8F80 2063                    TON2   RSTR   <SAVMAJ,=Z'0002'
0AFE  0002
0AFF  8382                                JMP    $B2
                                   *
                                   *
                                   * PUT XMIT IN FLAG IDLE, TURN ON RCV, DUMMY RCV, GET NO ILS RUPT
                                   *
0B00  BF00 1166                    TON3   STR    $R3,<CHAN
0B02  3042                                SOR    $R3,2                GET LA NUMBER
0B03  9830 11B9                           LDR    $R1,<ATLT.$R3        GET ID
0B05  8752                                CL     =$R2
0B06  9970 21F6                           CMR    $R1,=Z'21F6'
0B08  0981 0003                           BNE    TON10                B = BH4DLE
0B0A  A870 00EB                           LDR    $R2,=X'EB'
0B0C  AF00 0B9C                    TON10  STR    $R2,<DSC1
0B0E  B800 1166                           LDR    $R3,<CHAN            RESTORE CHANNEL NUMB
0B10  9870 0020                           LDR    $R1,=X'20'           INITIAL FLAG IDLE
0B12  9780 0BFA                           STH    $R1,<IFM
```

```
0B14  1C08                LDV    $R1,=8
0B15  9780 0BFB           STH    $R1,<XTEMP            SET FLAG TO READ XMIT FW REV
0B17  B380 0BA3           LNJ    $B3,<LOAD             LOAD CHANNEL PROGRAM AND EXECUTE
                    *
                    * READ P VALUE FOR RCV
                    *
0B19  C380 0F6B           LNJ    $B4,<RPVLU            READ P TO R5
0B1B  E870 021A           LDR    $R6,=X'21A'           SB
0B1D  E955                CMR    $R6,=$R5
0B1E  0903                BE     >$+3
0B1F  E3C0 048E           LNJ    $B6,ERRDB             P VALUE WRONG, AFTER START IO
                    *
                    * READ P VALUE FOR XMIT
                    *
0B21  8AD3                INC    =$R3                  GET XMIT CHANNEL
0B22  C380 0F6B           LNJ    $B4,<RPVLU
0B24  E870 0441           LDR    $R6,=X'441'           SB
0B26  E955                CMR    $R6,=$R5
0B27  0903                BE     >$+3
0B28  E3C0 0485           LNJ    $B6,ERRDB             WRONG P VALUE AFTER START, XMIT
                    *
                    * TURN ON TRANSMIT , BUT NOT RCV AND CHECK OPERATION
                    *
0B2A  8780 0BFA           CLH    <IFM                  SET FOR ABORT FILL
0B2C  8780 0BE6           CLH    <PRCV                 SET FOR RECEIVE START AT X'200'
0B2E  B380 0BA3           LNJ    $B3,<LOAD             LOAD CHANNEL PROGRAM AND EXECUTE
                    *
                    * READ XMIT STATUS, SHOULD BE COMPLETE
                    *
0B30  E870 1020           LDR    $R6,=X'1020'
0B32  8AD3                INC    =$R3                  GET BACK XMIT
0B33  C380 10D9           LNJ    $B4,<INXT             INPUT NEXT STATUS TO R5
0B35  E955                CMR    $R6,=$R5
0B36  0903                BE     >$+3
0B37  E3C0 0476           LNJ    $B6,ERRDB             WRONG XMIT STATUS
                    *
                    *
                    * CHECK P VALUE FOR RCV - SHOULD HAVE ILS ONLY
                    *
0B39  88D3                DEC    =$R3                  GET BACK RECEIVE CHANNEL
0B3A  C380 0F6B           LNJ    $B4,<RPVLU            READ P VALUE TO R5
0B3C  E870 0270           LDR    $R6,=X'270'           SB
0B3E  E955                CMR    $R6,=$R5
0B3F  0903                BE     >$+3
0B40  E3C0 046D           LNJ    $B6,ERRDB             INCORRECT RCV OPERATION
                    *
                    * CHECK P VALUE FOR XMIT, SHOULD HAVE COMPLETED
                    *
0B42  8AD3                INC    =$R3
0B43  C380 0F6B           LNJ    $B4,<RPVLU            READ P TO R5
0B45  E870 0403           LDR    $R6,=X'403'           SB
0B47  E955                CMR    $R6,=$R5
0B48  0903                BE     >$+3
0B49  E3C0 0464           LNJ    $B6,ERRDB             XMIT CHANNEL PROG DIDN'T COMPLETE
                    *
                    *
                    * IF FIRST PASS, LOOP, PRINT BBHDLC FIRMWARE REV NUMBER
                    *
0B4B  8980 1179           CMZ    <LOOP
0B4D  0996                BNE    >TON6                 BRANCH IF NOT FIRST LOOP
0B4E  8980 1182           CMZ    <PASSC
0B50  0993                BNE    >TON6                 BRANCH IF NOT FIRST PASS
                    *
                          CALL   ZV$TC,MESG4           PRINT BHCLA LINE
0B51  FBC0 0003
0B53  D380 0000      X
0B55  0F80
0B56  0B92
0B57  BF52                STR    $R3,=$R2
0B58  2041                SOR    $R2,1                 GET LINE NUMBER
0B59  AF00 1177           STR    $R2,<TPR
                          CALL   ZV$TD,TPR             TYPE LINE NUMBER
0B5B  FBC0 0003
0B5D  D380 0000      X
0B5F  0F80
0B60  1177
0B61  E380 0C9A           LNJ    $B6,<PREV             PRINT REV NUMBER
0B63  B800 1166     TON6  LDR    $R3,<CHAN             GET TESTED <CHANNEL
                    *
                    * 1.  ATTEMPT TO TRANSMIT WITH NO REQUEST TO SEND, TEST AND DIRECT CONNECT.
                    * 2.  ATTEMPT TO TRANSMIT WITH NO DIRECT CONNECT OR TEST MODE
                    * 3.  ATTEMPT TO TRANSMIT WITH BIT 2 OF LR2 = 1. (SHOULD INHIBIT CLOCK)
                    *     PART 3 IS DONE FOR BH4DLD ONLY.
                    *
                    *
                    *
                    *
0B65  8700 0B99           CL     <RTSFLG               CLEAR FLAG
0B67  9800 1185           LDR    $R1,<CBLOOP
0B69  1900 0B8A           BEZ    $R1,<TON9             IF NOT CONN. LOOP, BRANCH
0B6B  8752                CL     =$R2
0B6C  9B80 0B9F           LAB    $B1,<PSB              ADDRESS OF P SHOULD BE TABLE
                    *
0B6E  9820 0B9A     TON8  LDR    $R1,<DSC.$R2          GET  VALUE OF LR2
                    *
0B70  1900 0B8A           BEZ    $R1,<TON9             B = DONE
                    *
0B72  9780 0BF1           STH    $R1,<LR2CFG
0B74  8A80 0B99           INC    <RTSFLG               FLAG TO INHIBIT RTS, DIR CONN
0B76  B380 0BA3           LNJ    $B3,<LOAD             LOAD AND EXECUTE
                    *
                    * READ P FOR RECEIVE
                    *
0B78  C380 0F6B           LNJ    $B4,<RPVLU
0B7A  E870 0270           LDR    $R6,=X'270'           SB
0B7C  E955                CMR    $R6,=$R5
0B7D  0903                BE     >$+3
0B7E  E3C0 042F           LNJ    $B6,ERRDB             INVALID RCV REQUEST
                    *
                    * READ P FOR TRANSMIT
                    *
0B80  8AD3                INC    =$R3                  GET TRANSMIT CHANNEL
0B81  C380 0F6B           LNJ    $B4,<RPVLU            INPUT P VALUE
0B83  E86D                LDR    $R6,$B1.+$R2          GET SHOULD BE
```

```
0B84  0956                    CMR    $R5,=$R6
0B85  0283                    BGE    >$+3
0B86  E3C0 0427               LNJ    $B6,ERRDB          MORE THAN ONE XMIT REQUEST.
                      *
0B88  0F80 0B6E               B      <TON8
                      *
0B8A  B800 1166       TON9    LDR    $R3,<CHAN
0B8C  3E04                    ADV    $R3,=4             BUMP LINE NUMBER
0B8D  1C45                    LDV    $R1,=X'45'
0B8E  9780 0BF1               STH    $R1,<LR2CFG        RESTORE CONFIGURATION
0B90  0F80 0AF8               B      <TON1              DO FOR NEXT HDLC LINE
0B92  4843 4C41 2020  MESG4   TEXT   'HCLA  LINE  $'
0B95  4C49 4E45 2020
      2400
0B99  0000            RTSFLG  DC     0                  FLAG
                      *
                      * LR2 VALUES TO OUTPUT
                      *
0B9A  008B            DSC     DC     Z'008B'
0B9B  00C2                    DC     Z'00C2'
0B9C  00EB            DSC1    DC     Z'00EB'            0 IF BH4DLE
0B9D  0000                    RESV   2,0
                      *
                      * P VALUE FOR XMIT
                      *
0B9F  0425            PSB     DC     Z'0425'
0BA0  0425                    DC     Z'0425'
0BA1  0425                    DC     Z'0425'
0BA2  0000                    DC     0
                      *------------------------------------------------------------------
                      *
                      * SUBROUTINE TO SET UP CCB'S, LOAD CHANNEL PROGRAM, AND START EXECUTION
                      *
0BA3  8F00 2073       LOAD    SAVE   <SAV,=Z'2052'      SAVE B3,B6,R2,B1
0BA5  2052
                      *
                      * FORM  MASK OF ALL ONES EQUIVILENT TO CHAR SIZE
                      *
0BA6  8752                    CL     =$R2
0BA7  9800 1174               LDR    $R1,<HEAD          GET FRAME CONFIG INFO
0BA9  1806                    BLZ    $R1,>FMASK         BRANCH IF SUPERVISORY CONTROL BIT SET
0BAA  9800 116A               LDR    $R1,<CHSZ          GET CHAR SIZE
0BAC  9B80 0C91               LAB    $B1,<MSKTBL        GET ADDRESS OF TABLE OF MASKS
0BAE  A811                    LDR    $R2,$B1.$R1        GET MASK
                      *
0BAF  2008            FMASK   SOL    $R2,8
0BB0  A470 003D               OR     $R2,=X'3D'
0BB2  AF40 0041               STR    $R2,XTMSK          STORE MASK WORD
0BB4  0F00 0BBB               NOP    <LPT6
                      *
0BB6  9800 0C1B               LDR    $R1,<XRNG1
0BB8  8752                    CL     =$R2
0BB9  1B02                    BEVN   $R1,>LPT6
0BBA  8AD2                    INC    =$R2
0BBB  AF00 119B       LPT6    STR    $R2,<ODDFLG
0BBD  9A00 0C29               ADD    $R1,<XRNG2         TOTAL RANGE IN BYTES
0BBF  1041                    SOR    $R1,1              FORM RANGE IN WORDS
0BC0  9F00 116C               STR    $R1,<RANGE
                      *
                      * MODIFY CONTROL IF CONNECTOR OR EXTERNAL LOOP
                      *
0BC2  9800 1185               LDR    $R1,<CBLOOP        GET CONNECTOR LOOP FLAG
0BC4  1900 0BD1               BEZ    $R1,<LCT7
0BC6  9800 0B99               LDR    $R1,<RTSFLG
0BC8  1989                    BNEZ   $R1,>LCT7          BRANCH FOR NO RTS, DIR. CONN.
0BC9  9800 0BF1               LDR    $R1,<LR2CFG        GET LR2 INFO
0BCB  9570 FBFF               AND    $R1,=Z'FBFF'       STRIP OFF TEST
0BCD  9470 4800               OR     $R1,=Z'4800'       SET DIRECT CONNECT ,RTS
0BCF  9F00 0BF1               STR    $R1,<LR2CFG
0BD1  8700 0B99       LCT7    CL     <RTSFLG
0BD3  9800 1184               LDR    $R1,<XLOOP         GET EXTERNAL LOOP FLAG
0BD5  1900 0BDD               BEZ    $R1,<LPT3
0BD7  9800 0BF1               LDR    $R1,<LR2CFG
0BD9  9570 F3FF               AND    $R1,=Z'F3FF'       STRIP OFF TEST, DIRECT CONNECT
0BDB  9F00 0BF1               STR    $R1,<LR2CFG
                      *
0BDD  B800 1166       LPT3    LDR    $R3,<CHAN
0BDF  C380 0EEC               LNJ    $B4,<GENITZ        INITIALIZE
                      *
                      *
                      * SEND OUT LCT
                      *
0BE1  B380 10EB               LNJ    $B3,<SETLCT
0BE3  0BE5                    DC     <LCT1              LCT TABLE FOR THIS TEST
0BE4  0FA7                    B      >TST1A
                      *
                      * LCT TABLE
                      *
0BE5  0206            LCT1    DC     Z'0206'            P, MSB
0BE6  0007            PRCV    DC     Z'0007'            P, LSB
0BE7  0426                    DC     Z'0426'            P, MSB, XMIT
0BE8  0027                    DC     Z'0027'            P, LSB, XMIT
0BE9  000C                    DC     Z'000C'            CHANNEL
0BEA  000D                    DC     Z'000D'            LEVEL
0BEB  002C                    DC     Z'002C'            CHANNEL
0BEC  002D                    DC     Z'002D'            LEVEL
0BED  0023                    DC     Z'0023'            CRC RESIDUE 1, XMIT
0BEE  0024                    DC     Z'0024'            CRC RESIDUE 2, XMIT
                      *
                      * CONFIGURABLE PARAMETERS
                      *
0BEF  0003            PARBYT  DC     X'0003'            BYTE SIZE FOR LAST BYTE XMIT
0BF0  0004            RCVRES  DC     X'0004'            RESIDUE FOR LAST BYTE, RCV
0BF1  C714            LR2CFG  DC     Z'C714'            LR2
0BF2  0017            LR6RCV  DC     Z'0017'            CONFIG (LR6)
0BF3  0037            LR7XMT  DC     Z'0037'            XMIT CONFIG (LR7)
0BF4  FF3D            XTMSK   DC     Z'FF3D'            WORD SIZE MASK
0BF5  003C            TACC    DC     Z'003C'            XMIT ACTION CHAR COUNT
0BF6  0019            UFMCNT  DC     X'0019'            ACTION FRAME COUNTER - XMIT
0BF7  003E            ACTFLG  DC     Z'003E'            ACTION FLAG
0BF8  001D            RAC     DC     Z'001D'            RECEIVE ACTION CHAR COUNT
0BF9  001B            FRMCNT  DC     Z'001B'            FRAME COUNT, RCV
0BFA  003F            IFM     DC     Z'003F'            FILL STATE, BIT 0 = IFM, BIT 1 = INITIAL
```

```
                          *                          BITS 2 - 7 = RCV ACTION  FRAME COUNT
0BFB  0039                XTEMP   DC      Z'0039'          INITIALLY FLAG TO READ XMIT REV, 0 = DON'T.
                          *                              AFTERWARDS USED AS XMIT TEMP LOCATION
                          *
                          * CHANNEL PROGRAM WORK LOCATIONS
                          *
0BFC  0018                        DC      Z'0018'          CHAR COUNTER, RCV
0BFD  0038                        DC      Z'0038'          CHAR COUNTER, XMIT
0BFE  000E                        DC      Z'000E'          LR5 STORAGE, RCV
0BFF  002E                        DC      Z'002E'          LR5 STORAGE, XMIT
0C00  001A                        DC      Z'001A'          LR7, RCV
0C01  001E                        DC      Z'001E'          XMIT FW REV STORAGE
0C02  001F                        DC      Z'001F'          RCV FW REV STORAGE
                          *
0C03  003B                        DC      Z'003B'          XMIT FRAME COUNT
0C04  001C                        DC      Z'001C'          REC TEMP LOC
0C05  003A                        DC      Z'003A'          ERROR CODE
                          * SPARE
0C06  0002                        DC      Z'0002'
0C07  0022                        DC      Z'0022'
                          *
0C08  0000                        RESV    3,0
                          *
                          * SEND OUT CHANNEL PROGRAM
                          *
0C0B  8AD3                TST1A   INC     =$R3             GET XMIT CHANNEL
0C0C  9380 1095                   LNJ     $B1,<SDATA       SEND DATA
0C0E  11FF                        DC      <CCP1            RCV CHANNEL PROGRAM
0C0F  01AA                        DC      (CCP2-CCP1)*2
0C10  0200                        DC      X'200'           RAM ADDRESS
0C11  0000                        DC      0                EVEN CPU ADDRESS
                          *
                          * SEND OUT XMIT CHANNEL PROGRAM
                          *
0C12  9380 1095                   LNJ     $B1,<SDATA
0C14  12D4                        DC      <CCP2            XMIT CHANNEL PROGRAM
0C15  018A                        DC      (CCP3-CCP2)*2    RANGE
0C16  0400                        DC      X'400'           RAM ADDRESS
0C17  0000                        DC      0                EVEN BYTE ADDRESS
                          *
                          *
                          * SET UP CCB'S FOR DATA XFER
                          *
0C18  C380 10FF                   LNJ     $B4,<MCCB        MAKE CCB  FOR XMIT
0C1A  1160                XADD1   DC      <DUMMY           DATA ADDRESS
0C1B  0000                XRNG1   DC      0                RANGE
0C1C  0040                XCON1   DC      Z'0040'          CCB CONTROL WORD
                          *
0C1D  8280 0C1C                   LB      <XCON1,=Z'0020'  GET LAST BIT
0C1F  0020
0C20  0523                TESTT   BBT     >TESTM           BRANCH IF SET
                          *
                          * GIVE SECOND CCB
                          *
0C21  9800 0C1C                   LDR     $R1,<XCON1       GET CONTROL WORD
0C23  82D1                        LB      =$R1,=Z'0020'
0C24  0020
0C25  051E                        BBT     >TESTM
0C26  C380 10FF                   LNJ     $B4,<MCCB        MAKE CCB
0C28  1160                XADD2   DC      <DUMMY           XADDRESS
0C29  0000                XRNG2   DC      0                RANGE
0C2A  0060                XCON2   DC      X'60'            CONTROL
                          *
0C2B  8708 0C28                   CL      *<XADD2          CLEAR ADDRESS + CONTROL
                          *
0C2D  8280 0C2A                   LB      <XCON2,=Z'0020'  GET LAST BLOCK BIT
0C2F  0020
0C30  0513                        BBT     >TESTM           BRANCH IF SET
                          *
                          * GIVE 3D CCB
                          *
0C31  C380 10FF                   LNJ     $B4,<MCCB        MAKE CCB
0C33  1160                XADD3   DC      <DUMMY
0C34  0000                XRNG3   DC      0
0C35  0060                XCON3   DC      X'60'
                          *
0C36  8708 0C33                   CL      *<XADD3
0C38  8280 0C35                   LB      <XCON3,=Z'0020'  GET LAST BLOCK BIT
0C3A  0020
0C3B  0508                        BBT     >TESTM           BRANCH IF LAST BLOCK
                          *
                          * GIVE 4TH CCB
                          *
0C3C  C380 10FF                   LNJ     $B4,<MCCB
0C3E  1160                XADD4   DC      <DUMMY
0C3F  0000                XRNG4   DC      0
0C40  0060                        DC      X'60'            LAST, VALID
                          *
0C41  8708 0C3E                   CL      *<XADD4          CLEAR ADDRESS AND CONTROL
0C43  88D3                TESTM   DEC     =$R3             GET RCV CHAN BACK
                          *
                          *  SET UP ONE RCV CCB FOR EACH
                          * XMIT CCB.  A RCV CCB IS X'400'  HIGHER IN MEMORY THEN CORRESPONDING
                          * XMIT CCB.
                          *
0C44  C380 10FF                   LNJ     $B4,<MCCB        MAKE CCB
0C46  1C00                RADD1   DC      <RTB             RECEIVE BUFFER
0C47  0000                RRNG1   DC      0                RANGE
0C48  0040                RCON1   DC      X'40'            VALID
0C49  8280 0C48                   LB      <RCON1,=Z'0020'  GET LAST BIT
0C4B  0020
0C4C  0518                        BBT     >TESTY           BRANCH IF SET
                          * 2ND RECEIVE CCB
0C4D  C380 10FF                   LNJ     $B4,<MCCB        MAKE CCB
0C4F  1C00                RADD2   DC      <RTB             ADDRESS
0C50  0000                RRNG2   DC      0                RANGE
0C51  0040                RCON2   DC      X'40'            VALID
0C52  8280 0C51                   LB      <RCON2,=Z'0020'  GET LAST BIT
0C54  0020
0C55  050F                        BBT     >TESTY           BRANCH IF SET
                          * THIRD CCB
0C56  C380 10FF                   LNJ     $B4,<MCCB
0C58  1C00                RADD3   DC      <RTB
```

```
0C59  0000                  RRNG3  DC     0
0C5A  0040                  RCON3  DC     X'40'                  VALID
0C5B  8280 0C5A                    LB     <RCON3,=Z'0020'        GET LAST BIT
0C5D  0020
0C5E  0506                         BBT    >TESTY                 BRANCH IF SET
                            * MAKE 4TH  CCB
0C5F  C380 10FF                    LNJ    $B4,<MCCB
0C61  1C00                  RADD4  DC     <RTB
0C62  0000                  RRNG4  DC     0
0C63  0060                  RCON4  DC     X'60'                  LAST, VALID
                            *
                            *
                            * FILL RECEIVE WITH DEFAULT
0C64  C380 0F02             TESTY  LNJ    $B4,<FDFLT
                            *
                            *
                            * SHORTEN TIMEOUT FOR ABORT CASES
                            *
0C66  9870 01E0                    LDR    $R1,=480
0C68  A870 0006                    LDR    $R2,=6
0C6A  A900 116C                    CMR    $R2,<RANGE
0C6C  0203                         BL     >TESTV                 BRANCH IF RANGE >6
0C6D  9870 0018                    LDR    $R1,=24                200 MS DELAY
0C6F  9F00 0C8C             TESTV  STR    $R1,<TIMOUT
                            *
                            * START IO TO START XMIT CHANNEL PROG. INSURES
                            * START CONDITION IS IN DESIRED STATE
                            *
0C71  8AD3                         INC    =$R3
0C72  C380 0F0B             TESTU  LNJ    $B4,<CHCT              DO CHANNEL CONTROL
0C74  0F82                         B      >$+2
0C75  4000                         DC     Z'4000'                START IO
                            *
0C76  88D3                         DEC    =$R3                   GET BACK RCV
                            *
                            * SYNCHRONIZE WITH RTC AND GIVE ENOUGH TIME FOR INITIAL ILS  CHAR'S TO OCCUR
                            *
0C77  8700 0000         X          CL     <ZHRTCI                RTC  RESET VALUE
0C79  1C06                         LDV    $R1,=6
0C7A  9F00 0000         X          STR    $R1,<ZHRTCL            LEVEL
0C7C  9F00 0000         X          STR    $R1,<ZHRTCC            RTC
0C7E  0004                         RTCN
0C7F  1C01                         LDV    $R1,=1
0C80  9900 0000         X   TESTX  CMR    $R1,<ZHRTCC
0C82  0981 FFFD                    BNE    TESTX                  BRANCH IF HASN'T CHANGED
0C84  0005                         RTCF                          SHUT OFF CLOCK
                            *
                            * START I/O TO START RCV CHANNEL PROGRAM
                            *
0C85  C380 0F0B                    LNJ    $B4,<CHCT              DO CHANNEL CONTROL
0C87  0F82                         B      >$+2
0C88  4000                         DC     Z'4000'                START I/O
                            *
0C89  C380 0F1D                    LNJ    $B4,<TEST              TEST FOR STATUS COMPLETE
0C8B  0F82                         B      >$+2
0C8C  01E0                  TIMOUT DC     480                    TIMEOUT AFTER 4 SEC OR 200 MS
0C8D  8F80 2073                    RSTR   <SAV,=Z'2052'
0C8F  2052
0C90  8383                         JMP    $B3
                            *
                            * MASK TABLE
                            *
0C91  0000                  MSKTBL DC     0                      MASK FOR 8 BITS
0C92  0000                         DC     0
0C93  0000                         DC     0
0C94  0000                         DC     0
0C95  0000                         DC     X'0'
0C96  00E0                         DC     X'E0'                  5 BIT
0C97  00C0                         DC     X'C0'                  MASK FOR 6 BITS
0C98  0080                         DC     X'80'                  MASK FOR 7 BITS
0C99  0000                         DC     0                      EXTRA
                            ******************************************************************
                            *
                            *
                            * SUBROUTINES ARE CODED HERE
                            *
                            *
                            *
                            *  READ AND PRINT FIRMWARE REV NO.
                            *
0C9A  8F00 11AE             PREV   SAVE   <TMPSTR,=Z'1002'       R3,B4
0C9C  1002
0C9D  3B00 0CA0                    BEVN   $R3,<PREV5             BRANCH IF RCV CHANNEL
0C9F  88D3                         DEC    =$R3
0CA0  8754                  PREV5  CL     =$R4
0CA1  C380 0F0B                    LNJ    $B4,<CHCT              DO CHANNEL CONTROL
0CA3  0F82                         B      >$+2
0CA4  0100                         DC     Z'0100'                CCB LIST RESET
0CA5  9800 11EA                    LDR    $R1,<ERMG+1
0CA7  9970 2A2A                    CMR    $R1,=A'**'             SEE IF MLCP REV
0CA9  0904                         BE     >PREV2                 BRANCH IF MLCP REV NO.
0CAA  BF54                         STR    $R3,=$R4
0CAB  4F20                         MLV    $R4,=X'20'             FORM OFFSET FOR CHANNEL
0CAC  0F83                         B      >PREV1
0CAD  C380 0EEC             PREV2  LNJ    $B4,<GENITZ            GENERAL INITIALIZE
0CAF  CF00 0CB5             PREV1  STR    $R4,<REVLOC
0CB1  9380 10BF                    LNJ    $B1,<RDATA             READ DATA
0CB3  1C00                         DC     <RTB                   TO HERE
0CB4  0020                         DC     X'20'                  RANGE
0CB5  0000                  REVLOC DC     0                      ADDRESS
0CB6  0000                         DC     0                      EVEN BYTE BOUNDRY
                            *
                                   CALL   ZV$T,FREV              PRINT FIRMWARE REV
0CB7  FBC0 0003
0CB9  D380 0000         X
0CBB  0F80
0CBC  0D2D
0CBD  BF00 1178                    STR    $R3,<TEMP
0CBF  3043                         SOR    $R3,3                  GET LINE NUMBER/4
0CC0  BB80 1800                    LAB    $B3,<SDB
0CC2  2CFF                         LDV    $R2,=-1
0CC3  9800 11EA                    LDR    $R1,<ERMG+1            GET TEST  LABLE
0CC5  9970 2A2A                    CMR    $R1,=A'**'
```

```
0CC7  091D                             BE     >PREV3               BRANCH IF MLCP REV NUMBER
0CC8  2CFE                             LDV    $R2,=-2
                                       CALL   ZV$T,RPRT            DISPLAY RCV =
0CC9  FBC0 0003
0CCB  D380 0000          X
0CCD  0F80
0CCE  11F3
0CCF  BB80 119E                        LAB    $B3,<RHCFW1
0CD1  9800 1C0F                        LDR    $R1,<RTB+15
0CD3  9F00 1C00                        STR    $R1,<RTB             STORE FW REV
0CD5  0F8F                             B      >PREV3
                                PREV4  CALL   ZV$T,XPRT            DISPLAY XMIT =
0CD6  FBC0 0003
0CD8  D380 0000          X
0CDA  0F80
0CDB  11EC
0CDC  9800 1C0F                        LDR    $R1,<RTB+15          GET FW REV
0CDE  1068                             SAR    $R1,8                ALIGN XMIT REV
0CDF  8251                             NEG    =$R1                 MAKE REV +
0CE0  BB80 119C                        LAB    $B3,<XHCFW1
0CE2  9F00 1C00                        STR    $R1,<RTB
                                *
0CE4  9800 1C00                 PREV3  LDR    $R1,<RTB             GET REV NUMBER
0CE6  9570 00FF                        AND    $R1,=X'FF'
0CE8  9F33                             STR    $R1,$B3.$R3          STORE REV NUMB
0CE9  9570 00E0                        AND    $R1,=Z'00E0'
0CEB  1045                             SOR    $R1,5                ALIGN
0CEC  9F00 1177                        STR    $R1,<TPR
0CEE  1900 0CFC                        BEZ    $R1,<PREV6           IF NO UPPER BIT BRANCH
                                       CALL   ZV$ID,TPR            TYPE
0CF0  FBC0 0003
0CF2  D380 0000          X
0CF4  0F80
0CF5  1177
                                       CALL   ZV$T,DASH            TYPE DASH
0CF6  FBC0 0003
0CF8  D380 0000          X
0CFA  0F80
0CFB  0D3B
                                *
0CFC  9800 1C00                 PREV6  LDR    $R1,<RTB             GET REV NUMB
0CFE  9570 001F                        AND    $R1,=Z'001F'         STRIP TO LOW 5 BITS
0D00  9F00 1177                        STR    $R1,<TPR             PUT BACK
                                       CALL   ZV$ID,TPR            PRINT NUMBER
0D02  FBC0 0003
0D04  D380 0000          X
0D06  0F80
0D07  1177
0D08  2780 0CD6                 SIT13  BINC   $R2,<PREV4           GO BACK AND DO FOR XMIT
                                *
                                * IF PRINTING FIRMWARE REV, ALSO PRINT DATA SET STATUS
                                *
0D0A  B800 1178                        LDR    $R3,<TEMP            GET BACK CHANNEL
0D0C  9800 11EA                        LDR    $R1,<ERMG+1          GET TEST LABLE
0D0E  9970 2A2A                        CMR    $R1,=A'**'
0D10  0913                             BE     >SIT14               BRANCH IF MLCP REV NO.
0D11  9800 1C07                        LDR    $R1,<RTB+7           GET STATUS
0D13  1048                             SOR    $R1,8                GET STATUS
0D14  9F00 1177                        STR    $R1,<TPR
                                       CALL   ZV$TC,STPT           PRINT DATA SET STATUS
0D16  FBC0 0003
0D18  D380 0000          X
0D1A  0F80
0D1B  0D32
                                       CALL   ZV$THZ,TPR           PRINT VALUE
0D1C  FBC0 0003
0D1E  D380 0000          X
0D20  0F80
0D21  1177
0D22  0F87                             B      >SIT12
0D23  9800 1C00                 SIT14  LDR    $R1,<RTB             GET REV
0D25  9570 00FF                        AND    $R1,=Z'00FF'         STRIP
0D27  9F00 119A                        STR    $R1,<MLC-FR          STORE MLC FIRMWARE REV NUMBER
                                *
                                *
0D29  8F80 11AE                 SIT12  RSTR   <TMPSTR,=Z'1002'
0D2B  1002
0D2C  8386                             JMP    $B6                  RETURN
                                *
0D2D  2020 4657 2052            FREV   TEXT   '  FW REV$'
0D30  4556 2400
0D32  4441 5441 2053            STPT   TEXT   'DATA SET STATUS =$'
0D35  4554 2053 5441
      5455 5320 3D24
0D3B  202D 2024                 DASH   TEXT   ' - $'
                                *
                                *
                                *
                                * FILL ASCENDING DATA  LNJ $B3,<FACDTA
                                *
0D3D  8F00 202D                 FACDTA SAVE   <SAV2,=Z'6040'       R1,R2,B1
0D3F  6040
0D40  8751                             CL     =$R1
0D41  9B80 1800                        LAB    $B1,<SDB
0D43  8752                             CL     =$R2
0D44  A7DD                      FACDT1 STH    $R2,$B1.+$R1         FILL BUFFER
0D45  8AD2                             INC    =$R2
0D46  9970 0400                        CMR    $R1,=X'400'
0D48  09FC                             BNE    >FACDT1
0D49  8F80 202D                        RSTR   <SAV2,=Z'6040'
0D4B  6040
0D4C  8383                             JMP    $B3
                                *
                                * GENERATE CRC FOR SEND BUFFER
                                *
                                *
0D4D  1C08                      GCRC   LDV    $R1,=8               CHAR SIZE IS 8 FOR CRC TEST
0D4E  7CFF                             LDV    $R7,=-1
0D4F  8753                             CL     =$R3                 RANGE COUNTER
                                * * BEGINNING OF CHAR LOOP
0D50  9F52                      CHLUP  STR    $R1,=$R2
0D51  D2B0 1800                        LLH    $R5,<SDB.$R3         GET BYTE
0D53  8AD3                             INC    =$R3
```

```
                           *
                           * LOAD BYTE SIZE FOR LAST BYTE
                           *
0D54   3D1F                        CMV    $R3,=X'1F'
0D55   098C                        BNE    >GRC1               BRANCH IF NOT LAST BYTE
0D56   9800 089E                   LDR    $R1,<PCRC4
0D58   9570 000E                   AND    $R1,=X'E'           STRIP TO LAST BYTE INFO
0D5A   1041                        SOR    $R1,1
0D5B   A810 04AD                   LDR    $R2,<TCBBSZ.$R1     CONVERT BYTE CODE TO HEX
0D5D   2984                        BNEZ   $R2,>GRC1
0D5E   2C08                        LDV    $R2,=8              SET FOR 8 BITS
0D5F   0F00 0D50                   NOP    <CHLUP
                           *
0D61   3D20                GRC1    CMV    $R3,=X'20'
0D62   0280 0D6F                   BGE    <RCEND
                           *
0D64   2700 0D67           BITLUP  BDEC   $R2,<GRC2
0D66   0FEA                        B      >CHLUP              DO NEXT CHAR
0D67   E855                GRC2    LDR    $R6,=$R5
0D68   E657                        XOR    $R6,=$R7            R7 HAS RESIDUE
                           *
0D69   5041                        SOR    $R5,1               SHIFT CHAR
0D6A   7041                        SOR    $R7,1               SHIFT RESIDUE
0D6B   6B79                        BEVN   $R6,>BITLUP
0D6C   F670 8408                   XOR    $R7,=Z'8408'        OR WITH CCITT POLYNOMIAL
0D6E   0FF6                        B      >BITLUP
                           *
0D6F   8657                RCEND   CPL    =$R7                COMPLEMENT CRC
0D70   1C01                        LDV    $R1,=1
0D71   B857                        LDR    $R3,=$R7
0D72   B790 180F                   STH    $R3,<SDB+X'F'.$R1   STORE IN BUFFER
0D74   3048                        SOR    $R3,8               GET LEFT HALF OF CRC
0D75   B780 1810                   STH    $R3,<SDB+X'10'      STORE
0D77   B870 00E2                   LDR    $R3,=Z'00E2'
0D79   B790 1810                   STH    $R3,<SDB+X'10'.$R1
0D7B   B870 00F0                   LDR    $R3,=X'00F0'
0D7D   B780 1811                   STH    $R3,<SDB+X'11'
0D7F   8384                        JMP    $B4
                           *
                           * DATA HEADER ROUTINE
                           *
                           *   THIS SUBROUTINE FORMATS THE HEADER ON EACH FRAME BASED
                           *   ON THE FOLLOWING INPUTS:
                           *
                           *        TCB, AFX, CFX, + LCF FOR XMIT LR7
                           *        FLAG HEAD WHICH IS INTERPRETED AS FOLLOWS
                           *
                           *             BIT 0 = 1 MEANS SUPERVISORY CONTROL BIT SET
                           *             BITS 5-7 ARE THE TCB VALUE
                           *             BITS 8 - 11 ARE THE ADDRESS FIELD LENGTH
                           *             BITS 12-15 ARE THE LCF LENGTH
                           *
                           *
                           *   THE SUBROUTINE ALSO STRIPS XMIT DATA TO THE
                           *   BYTE SIZE SPECIFIED IN THE FOLLOWING PRIORITY:
                           *        SUPERVISORY BIT>TCB>LR7 BITS 5,6,7
                           *
0D80   8F00 2073           GHEAD   SAVE   <SAV,Z'FFFF'        SAVE ALL
0D82   FFFF
0D83   AB80 1800                   LAB    $B2,<SDB
0D85   DC80 0ECF                   LDB    $B5,<PARPTR         GET POINTER TO PARAMETER LIST
                           * ADDRESS FIELD
0D87   8700 116A                   CL     <CHSZ               CLEAR CHAR SIZE
0D89   1C01                        LDV    $R1,=1
0D8A   82C5 0001                   LB     $B5.1,=X'40'        GET AFX BIT
0D8C   0040
0D8D   0592                        BBF    >CFP                IF NO AFX, GO TO CONTROL FIELD
0D8E   C290 1174                   LLH    $R4,<HEAD.$R1       GET HEAD INFO
0D90   1C00                        LDV    $R1,=0
0D91   4044                        SOR    $R4,4               GET ADDRESS FIELD LENGTH
0D92   88D4                        DEC    =$R4
0D93   A092                ADPR    LDH    $R2,$B2.$R1         PICK UP BYTE
0D94   9954                        CMR    $R1,=$R4
0D95   0907                        BE     >EADF
                           *
0D96   8852                        LBF    =$R2,=1             SET CONTINUATION BIT =0
0D97   0001
0D98   8B52                        LBC    =$R2,=Z'0080'       COMPLEMENT MSB TO MARK 8 BIT DATA
0D99   0080
0D9A   A7DE                        STH    $R2,$B2.+$R1
0D9B   0FF8                        B      >ADPR
                           *
0D9C   8952                EADF    LBT    =$R2,=1             SET CONTINUATION BIT FOR LAST
0D9D   0001
0D9E   A7DE                        STH    $R2,$B2.+$R1
                           *
                           * CONTROL FIELD
0D9F   A092                CFP     LDH    $R2,$B2.$R1
0DA0   8852                        LBF    =$R2,Z'0001'
0DA1   0001
0DA2   A792                        STH    $R2,$B2.$R1         SET ITFF IND = 0
0DA3   8280 1174                   LB     <HEAD,=Z'8000'      GET CFX BIT
0DA5   8000
0DA6   0589                        BBF    >CFP1
0DA7   A470 0001                   OR     $R2,=1              SET ITFF IND = 1
0DA9   8B52                        LBC    =$R2,=Z'0080'       SET MSB TO MARK 8 BIT DATA
0DAA   0080
0DAB   A792                        STH    $R2,$B2.$R1
0DAC   1C08                        LDV    $R1,=8              BIT SET TO INDICATE AT 'DCHSZ' THAT
                           *                                  MASK IS 8 BIT
0DAD   9F00 116A                   STR    $R1,<CHSZ
                           *
0DAF   8AD1                CFP1    INC    =$R1                BUMP TO NEXT BYTE
0DB0   82C5 0001                   LB     $B5.1,=X'80'        GET CFX
0DB2   0080
0DB3   0585                        BBF    >TCBP
                           *
0DB4   A870 00FF                   LDR    $R2,=X'FF'
0DB6   A792                        STH    $R2,$B2.$R1         SET SECOND CONTROL WORD TO ALL ONES
0DB7   8AD1                        INC    =$R1                BUMP FOR CONTROL FIELD EXTENTION
                           *
                           * TCB BYTE PROCESSING
0DB8   82C5 0001           TCBP    LB     $B5.1,=X'20'        GET TCB BIT INFORMATION
```

```
0DBA  0020
0DBB  0580 0DCC            BBF    <LCFP                 B = NO TCB
                     *
0DBD  A280 1174            LLH    $R2,<HEAD             GET TCB VALUE
0DBF  A570 0007            AND    $R2,=7                STRIP
0DC1  2D03                 CMV    $R2,=3                CHECK FOR BIT INVERSION
0DC2  0907                 BE     >TCBP1                IF SO DON'T STORE CHAR SIZE
0DC3  8980 116A            CMZ    <CHSZ
0DC5  0980 0DC9            BNE    <TCBP1                B = CHAR SIZE ALREADY SET
0DC7  AF00 116A            STR    $R2,<CHSZ
                     *
0DC9  8952           TCBP1 LBT    =$R2,=X'80'           SET LAST BIT
0DCA  0080
0DCB  A792                 STH    $R2,$B2.$R1           PUT IN TCB
                     *
                     * LCF PROCESSING
0DCC  82C5 0001      LCFP  LB     $B5.1,=X'01'
0DCE  0001
0DCF  0580 0DE1            BBF    <DCHSZ                B = NO LCF
0DD1  C800 1174            LDR    $R4,<HEAD
0DD3  C570 000F            AND    $R4,=X'0F'
0DD5  2C01                 LDV    $R2,=1
                     *
0DD6  D292           LCFP1 LLH    $R5,$B2.$R1           GET BYTE
0DD7  A954                 CMR    $R2,=$R4              CHECK IF LAST
0DD8  0906                 BE     >ELCF
0DD9  8855                 LBF    =$R5,=X'80'
0DDA  0080
0DDB  D7DE                 STH    $R5,$B2.+$R1
0DDC  8AD2                 INC    =$R2
0DDD  0FF9                 B      >LCFP1
                     *
0DDE  8955           ELCF  LBT    =$R5,=X'80'
0DDF  0080
0DE0  D7DE                 STH    $R5,$B2.+$R1
                     *
                     * SET BYTE SIZE
                     *
0DE1  C800 116A      DCHSZ LDR    $R4,<CHSZ             GET CHAR SIZE
0DE3  4988                 BNEZ   $R4,>STDTA
0DE4  A845 0001            LDR    $R2,$B5.1             GET XMIT LR7
0DE6  A570 000E            AND    $R2,=X'E'             STRIP TO BYTE SIZE
0DE8  2041                 SOR    $R2,1
0DE9  C820 04AD            LDR    $R4,<TCBBSZ.$R2       PICK UP ACTUAL SIZE
0DEB  C570 0007      STDTA AND    $R4,=7
0DED  CF00 116A            STR    $R4,<CHSZ
                     * STRIP DATA TO SIZE
0DEF  A800 116A            LDR    $R2,<CHSZ
0DF1  D820 0E02            LDR    $R5,<DMASK.$R2        GET MASK
0DF3  DF00 1176            STR    $R5,<MASK
0DF5  A851                 LDR    $R2,=$R1              GET BYTE INDEX
                     *
0DF6  8754                 CL     =$R4
0DF7  C0A2           GDAT8 LDH    $R4,$B2.$R2           GET DATA
0DF8  C500 1176            AND    $R4,<MASK             STRIP TO SIZE
0DFA  C7EE                 STH    $R4,$B2.+$R2          STORE BACK
0DFB  A970 0800            CMR    $R2,=X'800'
0DFD  09FA                 BNE    >GDAT8                BRANCH IF MORE TO GO
0DFE  8F80 2073      GDAT7 RSTR   <SAV,Z'FFFF'
0E00  FFFF
0E01  8384                 JMP    $B4
                     *
                     * MASK TABLE
                     *
0E02  00FF           DMASK DC     X'FF'                 8 BITS
0E03  0001                 DC     1                     1 BIT
0E04  0003                 DC     3                     2 BITS
0E05  0007                 DC     7                     3 BITS
0E06  000F                 DC     X'F'                  4 BITS
0E07  001F                 DC     X'1F'                 5 BITS
0E08  003F                 DC     X'3F'                 6 BITS
0E09  007F                 DC     X'7F'                 7 BITS
                     *
                     *
                     * INVERT DATA ON A BYTE BY BYTE BASIS - ASSUMES LCF LENGTH =4
                     *   RANGE OF X'1F'
                     *
0E0A  8F00 2073      IVRT  SAVE   <SAV,=Z'FFFF'
0E0C  FFFF
0E0D  2C05                 LDV    $R2,=5                BYTE INDEX
0E0E  F800 089C            LDR    $R7,<PCRC2            GET RCV CONFIG
0E10  F570 0007            AND    $R7,=7                STRIP TO BYE SIZE
                     *
0E12  8700 1178      BYTTZ CL     <TEMP
0E14  9857                 LDR    $R1,=$R7              NEW BYTE SIZE
0E15  4C01                 LDV    $R4,=1
0E16  4000                 SOL    $R4,0                 PUT 1 IN MSB OF BYTE
0E17  1C01                 LDV    $R1,=1
0E18  8AD2                 INC    =$R2
0E19  D2A0 1800            LLH    $R5,<SDB.$R2          GET BYTE
0E1B  2D20                 CMV    $R2,=X'20'
0E1C  0205                 BL     >VERT
0E1D  8F80 2073            RSTR   <SAV,=Z'FFFF'
0E1F  FFFF
0E20  8384                 JMP    $B4                   ALL DONE
                     *
0E21  82D5           VERT  LB     =$R5,0                GET BIT LOW ORDER
0E22  0000
0E23  9E54                 SWR    $R1,=$R4              GET MASK FOR HIGH ORDER
0E24  8A00 1178            LBS    <TEMP,0               PUT IN INVERTED POSISTION
0E26  0000
0E27  9E54                 SWR    $R1,=$R4
0E28  8A00 1178            LBS    <TEMP,0               SET BIT, LOWER ORDER
0E2A  0000
0E2B  1001                 SOL    $R1,1
0E2C  4041                 SOR    $R4,1                 SET FOR NEXT BIT POSISTION
0E2D  9954                 CMR    $R1,=$R4
0E2E  0380 0E21            BLE    <VERT
0E30  0F80 0E12            B      <BYTTZ                DO NEXT BYTE
                     *
                     *
                     * PASS TEST PARAMETERS + FORMAT HEADER
                     *
```

```
0E32  AF80 0ECF     PASS    STB    $B2,<PARPTR          STORE POINTER TO PARAMETERS
0E34  C380 2000             LNJ    $B4,<PLB             PRINT LABLE
                    *
0E36  9872                  LDR    $R1,+$B2             PICK UP RECEIVE CONFIGURATION
0E37  1008                  SOL    $R1,8
0E38  9470 0017             OR     $R1,=X'17'
0E3A  9F00 0BF2             STR    $R1,<LR6RCV
0E3C  9872                  LDR    $R1,+$B2             PICK UP XMIT CONFIGURATION
0E3D  9F52                  STR    $R1,=$R2
0E3E  A570 000E             AND    $R2,=X'E'            XMIT CHAR SIZE
0E40  1008                  SOL    $R1,8
0E41  9470 0037             OR     $R1,=X'37'
0E43  9F00 0BF3             STR    $R1,<LR7XMT
0E45  9802                  LDR    $R1,$B2              GET LAST BYTE INFORMATION
0E46  9570 000F             AND    $R1,=X'F'
0E48  9780 0BF0     PASSD   STH    $R1,<RCVRES
0E4A  1008                  SOL    $R1,8
0E4B  9470 0003             OR     $R1,=3
0E4D  9F00 0BEF             STR    $R1,<PARBYT          PARTIAL BYTE INFORMATION
0E4F  9872          PASSA   LDR    $R1,+$B2             GET FILL MODE
0E50  1008                  SOL    $R1,8
0E51  9470 003F             OR     $R1,=X'3F'
0E53  9570 E0FF             AND    $R1,=Z'E0FF'         STRIP TO PERTINENT BITS
0E55  9F00 0BFA             STR    $R1,<IFM
0E57  9872                  LDR    $R1,+$B2             PICK UP LR2 CONTROL
0E58  1008                  SOL    $R1,8
0E59  9470 0014             OR     $R1,=X'14'
0E5B  9F00 0BF1             STR    $R1,<LR2CFG
                    *
0E5D  9872                  LDR    $R1,+$B2             GET ACTION FLAG
0E5E  9780 0BF7             STH    $R1,<ACTFLG
0E60  9082                  LDH    $R1,$B2              RCV FRAME
0E61  9570 003F             AND    $R1,=X'3F'
0E63  9480 0BFA             ORH    $R1,<IFM
0E65  9780 0BFA             STH    $R1,<IFM
0E67  9872                  LDR    $R1,+$B2             RCV ACTION CHAR COUNT
0E68  9780 0BF8             STH    $R1,<RAC
0E6A  9082                  LDH    $R1,$B2              XMIT ACTION RCV FRAME
0E6B  9780 0BF6             STH    $R1,<UFMCNT
0E6D  9872                  LDR    $R1,+$B2
0E6E  9780 0BF5             STH    $R1,<TACC            XMIT ACTION CHAR COUNT
0E70  9CF2                  LDB    $B1,+$B2             PICK UP ADDRESS
0E71  9F80 0C1A             STB    $B1,<XADD1
0E73  CBC1 0400             LAB    $B4,$B1.X'400'
0E75  CF80 0C46             STB    $B4,<RADD1
0E77  9872                  LDR    $R1,+$B2             PICK UP RANGE
0E78  9F00 0C1B             STR    $R1,<XRNG1
0E7A  A280 0BF7             LLH    $R2,<ACTFLG          GET FLAG
0E7C  82D2                  LB     =$R2,=Z'0008'
0E7D  0008
0E7E  0582                  BBF    >PASSB
0E7F  1E04                  ADV    $R1,=4               MAKE ROOM FOR CRC + RESIDUES
0E80  9F00 0C47     PASSB   STR    $R1,<RRNG1
                    *
0E82  8780 0BE6             CLH    <PRCV                RECEIVE P COUNT
0E84  8700 0C29             CL     <XRNG2
0E86  8700 0C28             CL     <XADD2
0E88  8700 0C2A             CL     <XCON2
0E8A  8700 0C35             CL     <XCON3
0E8C  8752                  CL     =$R2
0E8D  9872                  LDR    $R1,+$B2             PICK UP CCB CONTROL 1
0E8E  9F00 0C1C             STR    $R1,<XCON1
0E90  9F00 0C48             STR    $R1,<RCON1
0E92  82D1                  LB     =$R1,=Z'0020'        CHECK LAST BLOCK BIT
0E93  0020
0E94  0583                  BBF    >CCBA                BRANCH IF NOT LAST CCB
0E95  0F80 0ECE             B      <PASEND              DONE
                    *
                    *
0E97  9CF2          CCBA    LDB    $B1,+$B2             PICK UP ADDRESS
0E98  9F80 0C28             STB    $B1,<XADD2
0E9A  CBC1 0400             LAB    $B4,$B1.X'400'
0E9C  CF80 0C4F             STB    $B4,<RADD2
0E9E  9872                  LDR    $R1,+$B2             RANGE
0E9F  9F00 0C29             STR    $R1,<XRNG2
0EA1  9F00 0C50             STR    $R1,<RRNG2
0EA3  A872                  LDR    $R2,+$B2             CONTROL WORD
0EA4  AF00 0C2A             STR    $R2,<XCON2
0EA6  AF00 0C51             STR    $R2,<RCON2
0EA8  8754                  CL     =$R4
0EA9  82D2                  LB     =$R2,=Z'0020'
0EAA  0020
0EAB  0523                  BBT    >PASEND              BRANCH  IF LAST BLOCK
0EAC  9CF2                  LDB    $B1,+$B2
0EAD  9F80 0C33             STB    $B1,<XADD3
0EAF  CBC1 0400             LAB    $B4,$B1.X'400'       GET RECEIVE ADDRESS
0EB1  CF80 0C58             STB    $B4,<RADD3
0EB3  9872                  LDR    $R1,+$B2             RANGE
0EB4  9F00 0C34             STR    $R1,<XRNG3
0EB6  9F00 0C59             STR    $R1,<RRNG3
0EB8  9872                  LDR    $R1,+$B2
0EB9  9F00 0C35             STR    $R1,<XCON3
0EBB  9F00 0C5A             STR    $R1,<RCON3
0EBD  82D1                  LB     =$R1,=Z'0020'
0EBE  0020
0EBF  0500 0ECE             BBT    <PASEND              BRANCH IF LAST CCB
0EC1  9CF2                  LDB    $B1,+$B2
0EC2  9F80 0C3E             STB    $B1,<XADD4
0EC4  CBC1 0400             LAB    $B4,$B1.X'400'
0EC6  CF80 0C61             STB    $B4,<RADD4
0EC8  9872                  LDR    $R1,+$B2
0EC9  9F00 0C3F             STR    $R1,<XRNG4
0ECB  9F00 0C62             STR    $R1,<RRNG4
0ECD  9872                  LDR    $R1,+$B2             DUMMY TO BUMP RETURN
0ECE  8383          PASEND  JMP    $B3
                    *
0ECF  0000          PARPTR  RESV   $AF,0                CURRENT PARAMETER POINTER
                    *
                    *       FIND ACTIVE HDLC CHANNEL. START SEARCHING WITH
                    *       CHANNEL NUMBER CONTAINED IN R3.
                    *
                    *
                    *            LNJ   $B4,FLN
```

```
                           *         B       >RET      NO MORE LINES
                           *                              NORMAL RETURN
                           *
                           *
                           *
0ED0  8F40 114C            FLN       SAVE    SAV1,=Z'4020'          R1,B2
0ED2  4020
0ED3  3042                           SOR     $R3,2                  GO FROM CHANNEL TO ADAPTER NUMBER
0ED4  AB80 11B9                      LAB     $B2,<ATLT              CHANNEL TABLE
0ED6  3D04                 LN2       CMV     $R3,=X'4'
0ED7  030C                           BG      >LN3                   NO MORE LINES
0ED8  9830 11B9                      LDR     $R1,<ATLT,$R3
0EDA  9970 21F6                      CMR     $R1,=X'21F6'           CHECK IF ID FOR BBHDLC1
0EDC  0906                           BE      >LN4
0EDD  9970 21F7                      CMR     $R1,=X'21F7'
0EDF  0903                           BE      >LN4                   IS BBH4DLE
0EE0  8AD3                           INC     =$R3
0EE1  0FF5                           B       >LN2                   TRY NEXT LINE
0EE2  9874                 LN4       LDR     $R1,+$B4
0EE3  8FC0 1139            LN3       RSTR    SAV1,=Z'4020'
0EE5  4020
0EE6  3002                           SOL     $R3,2                  GO FROM ADAPT NUMBER TO CHANNEL
0EE7  8384                           JMP     $B4
                           *
                           *                CHANGE CHANNEL
                           *
                           *        $R3 -CONTAINS CHANNEL WANTED
                           *        $R4 -CONTAINS I/O CONTROL WORD TO BE CHANGED
                           *
                           *         LNJ       $B4,<CGSCH
                           *
                           *
0EE8  3006                 CGSCH     SOL     $R3,6                  SHIFT TO CHANNEL POSITION
0EE9  C453                           OR      $R4,=$R3               OR CHANNEL NUMBER INTO CONTROLWORD
0EEA  3046                           SOR     $R3,6                  SHIFT TO NORMAL POSITION
0EEB  8384                           JMP     $B4                    GO BACK
                           *
                           * GIVE MLCP GENERAL INITIALIZE
                           *
0EEC  8F00 202D            GENITZ    SAVE    <SAV2,=Z'0808'         B4
0EEE  0808
0EEF  C380 0ED0                      LNJ     $B4,<FLN
0EF1  0000                           HLT                            NO ACTIVE LINES
0EF2  C800 11AA                      LDR     $R4,<CONT9             GET FUN CODE
0EF4  C380 0EE8                      LNJ     $B4,<CGSCH             FORM IO CONTROL
0EF6  8070 8000                      IO      =Z'8000',=$R4          INITIALIZE
0EF8  0054
0EF9  0703                           BIOT    >ITZ
0EFA  E380 0FBA                      LNJ     $B6,<ERRMB
0EFC  C380 1067            ITZ       LNJ     $B4,<DLAYLG
                           *
0EFE  8F80 202D                      RSTR    <SAV2,=Z'0808'
0F00  0808
0F01  8384                           JMP     $B4
                           *
                           * FILL RECEIVE BUFFER WITH DEFAULT
                           *
                           FDFLT     CALL    ZV$F,RTB,DFLT,DRNG
0F02  FBC0 0003
0F04  D380 0000      X
0F06  0F80
0F07  1C00
0F08  1172
0F09  1173
0F0A  8384                           JMP     $B4
                           *
                           * OUTPUT CHANNEL CONTROL
                           *
                           *    LNJ  $B4,CHCT
                           *    B   >$+2             RETURN
                           *    DC    XX             XX = CHANNEL CONTROL
                           *
0F0B  8F00 202D            CHCT      SAVE    <SAV2,=Z'0C0D'         R4,R5,B5,B7,B4
0F0D  0C0D
0F0E  D874                           LDR     $R5,+$B4               DUMMY
0F0F  D874                           LDR     $R5,+$B4               GET CONTROL WORD
0F10  C800 11A8                      LDR     $R4,<CONT7             FUN CODE FOR CCB CONTROL
0F12  C380 0EE8                      LNJ     $B4,<CGSCH             FORM IO CONTROL WORD
0F14  8055                           IO      =$R5,=$R4              OUTPUT CCB CONTROL
0F15  0054
0F16  0703                           BIOT    >CHZ
0F17  E380 0FBA                      LNJ     $B6,<ERRMB             ERROR IO WAS NAK'ED
                           *
0F19  8F80 202D            CHZ       RSTR    <SAV2,=Z'0C0D'
0F1B  0C0D
0F1C  8384                           JMP     $B4
                           *
                           * WAIT FOR STATUS COMPLETE
                           *   LNJ  $B4,<TEST
                           *    B      $+2      RETURN
                           *    DC     XX       TIMEOUT IN 120TH'S SEC.
                           *
                           *
0F1D  8F00 202D            TEST      SAVE    <SAV2,=Z'4909'         R1,4,7,B4,7
0F1F  4909
0F20  DB80 0000      X               LAB     $B5,<ZHPFR             CLEAR B5
0F22  8751                           CL      =$R1
0F23  9F00 0000      X               STR     $R1,<ZHRTCI            ZERO OUT RTC RESET VALUE
0F25  1C01                           LDV     $R1,=1
0F26  9F00 0000      X               STR     $R1,<ZHRTCL            SET FOR RUPT LEVEL 1
0F28  9874                           LDR     $R1,+$B4               DUMMY IO INCREMENT B4
0F29  9874                           LDR     $R1,+$B4               PICK UP TIMEOUT VALUE
0F2A  9F00 0000      X               STR     $R1,<ZHRTCC            SET REAL TIME CLOCK
0F2C  9F00 1167                      STR     $R1,<ELPS
0F2E  0004                           RTCN
0F2F  C380 10D9                      LNJ     $B4,<INXT              INPUT NEXT STATUS
0F31  82D5                 TESTZ     LB      =$R5,=Z'1000'          TEST FOR STATUS COMPLETE
0F32  1000
0F33  0511                           BBT     >TESTZ1                BRANCH IF COMPLETE
0F34  8980 0000      X               CMZ     <ZHRTCC                TEST RTC
0F36  090E                           BE      >TESTZ1                TIMEOUT
                           *
0F37  C800 11A7                      LDR     $R4,<CONT6             FUNCTION CODE
```

```
0F39  C380 0EE8                 LNJ    $B4,<CGSCH
0F3B  8055                      IO     =$R5,=$R4           INPUT STATUS
0F3C  0054
0F3D  0774                      BIOT   >TESTZ              BRANCH MEANS TRY AGAIN
0F3E  E380 0FBA                 LNJ    $B6,<ERRMB          INPUT STATUS WAS NAK'ED
0F40  0000                      HLT
                          *
                          *
                          * INPUT RESIDUAL RANGE
                          *
0F41  8F00 202D           INRNG  SAVE   <SAV2,=Z'4909'
0F43  4909
0F44  9870 9999           TESTZ1 LDR    $R1,=Z'9999'        DEFAULT
0F46  9F00 1183                  STR    $R1,<RRNG           RESIDUAL RANGE
0F48  C800 11A3                  LDR    $R4,<CONT2          FC
0F4A  C380 0EE8                  LNJ    $B4,<CGSCH          OR IN CHANNEL
0F4C  8000 1183                  IO     <RRNG,=$R4          INPUT RANGE
0F4E  0054
0F4F  0700 0F53                  BIOT   <TESTZ2
0F51  E380 0FBA                  LNJ    $B6,<ERRMB          COMMAND WAS NAK'ED
0F53  9800 1167           TESTZ2 LDR    $R1,<ELPS           GET TIMEOUT VALUE
0F55  0005                       RTCF                       TURN OFF RTC
0F56  9200 0000       X          SUB    $R1,<ZHRTCC         SUBTRACT PRESENT
0F58  C380 0F5E                  LNJ    $B4,<UPTM           UPDATE TIME TOTAL
                          *
0F5A  8F80 202D                  RSTR   <SAV2,=Z'4909'
0F5C  4909
0F5D  8384                       JMP    $B4                 EXIT
                          *
                          * UPDATE TIME TOTAL
                          *
0F5E  9A00 1198           UPTM   ADD    $R1,<TTOT
0F60  9F00 1198                  STR    $R1,<TTOT
0F62  1041                       SOR    $R1,1
0F63  9900 1191                  CMR    $R1,<HRTZ           COMPARE WITH TICKS FOR  1 SEC
0F65  0205                       BL     >UPTM1
0F66  8700 1198                  CL     <TTOT               CLEAR TOTAL
0F68  8A80 1199                  INC    <SEC
0F6A  8384                UPTM1  JMP    $B4
                          *
                          * READ P VALUE + ERROR CODE
                          *
                          *      LNJ $B4,<RPVLU
                          *
0F6B  8F00 204D           RPVLU  SAVE   <SAV4,=Z'5858'      R1,3,4 B1,3,4
0F6D  5858
0F6E  1C3A                       LDV    $R1,=X'3A'
0F6F  9780 1055                  STH    $R1,<DMPLOC         SET TO READ ERROR CODE
0F71  B380 10EB                  LNJ    $B3,<SETLCT         SEND OUT BYTE DIRECTOR
0F73  1055                       DC     <DMPLOC
0F74  B380 108D                  LNJ    $B3,<INBYTE         INPUT BYTE
0F76  DF00 1162                  STR    $R5,<ERRCD          STORE
                          *
0F78  8756                       CL     =$R6
0F79  D956                       CMR    $R5,=$R6
0F7A  0903                       BE     >LPTS4
0F7B  E3C0 0032                  LNJ    $B6,ERRDB           FATAL ERROR DURING CCP
                          *
0F7D  9870 0006           LPTS4  LDR    $R1,=6
0F7F  3B03                       BEVN   $R3,>PVLU2          BRANCH IF READ
0F80  88D3                       DEC    =$R3                GET READ CHANNEL
0F81  1C26                       LDV    $R1,=38
0F82  9780 1055           PVLU2  STH    $R1,<DMPLOC
0F84  B380 10EB                  LNJ    $B3,<SETLCT         SEND BYTE DIRECTOR
0F86  1055                       DC     <DMPLOC
0F87  B380 108D                  LNJ    $B3,<INBYTE         INPUT BYTE
0F89  DF00 1163                  STR    $R5,<PVLU           LSB OF P
0F8B  8AD1                       INC    =$R1
0F8C  9780 1055                  STH    $R1,<DMPLOC
0F8E  B380 10EB                  LNJ    $B3,<SETLCT         SEND BYTE DIRECTOR
0F90  1055                       DC     <DMPLOC
                          *
0F91  B380 108D                  LNJ    $B3,<INBYTE         INPUT BYTE
0F93  5048                       SOR    $R5,8
0F94  D400 1163                  OR     $R5,<PVLU           FORM LSB + MSB
0F96  DF00 1163                  STR    $R5,<PVLU
                          *
0F98  D570 0FFF                  AND    $R5,=Z'0FFF'        STRIP TO 12 BITS
0F9A  8F80 204D                  RSTR   <SAV4,=Z'5858'
0F9C  5858
0F9D  8384                       JMP    $B4
                          *
                          *
                          *
                          * READ DATA SET STATUS
0F9E  8F40 109E           DSSTA  SAVE   SAV3,=Z'0008'       SAVE B4
0FA0  0008
0FA1  C800 11AD                  LDR    $R4,<CONT12         GET FUNCTION CODE
0FA3  C380 0EE8                  LNJ    $B4,<CGSCH          OR IN CHANNEL NUMBER
0FA5  8055                       IO     =$R5,=$R4           INPUT DATA SET STATUS
0FA6  0054
0FA7  0703                       BIOT   >$+2+$AF
0FA8  E380 0FBA                  LNJ    $B6,<ERRMB          COMMAND WAS NAK'ED
0FAA  8FC0 1092                  RSTR   SAV3,=Z'0008'
0FAC  0008
0FAD  8384                       JMP    $B4                 DONE
                          *
                          * ERROR ROUTINES
                          *
0FAE  8F00 2053           ERRDB  SAVE   <SAV5,=Z'FFFF'
0FB0  FFFF
0FB1  9870 4442                  LDR    $R1,=A'DB'
0FB3  9F00 11E9                  STR    $R1,<ERMG           SET UP ERR MESSAGE FOR DAUGHT. BOATER
0FB5  9853                       LDR    $R1,=$R3            GET CHANNEL NUMBER
0FB6  1041                       SOR    $R1,1               CONVERT TO LINE
0FB7  9F00 1165                  STR    $R1,<LNBM
0FB9  0F8D                       B      >ERROR
                          *
0FBA  8F00 2053           ERRMB  SAVE   <SAV5,=Z'FFFF'
0FBC  FFFF
0FBD  9870 4D42                  LDR    $R1,=A'MB'
0FBF  9F00 11E9                  STR    $R1,<ERMG           SET ERROR MESSAGE FOR MOTHER
0FC1  1CFF                       LDV    $R1,=-1
```

```
0FC2  9F00 1165                       STR    $R1,<LNBM            SET LINE NUMBER TO NULL
0FC4  0F80 0FC6                       B      <ERROR               GO REPORT ERROR
                               *
                               *      EERROR PRINT ROUTINE
                               *
                               *EELNJ  $B7,<ERROR
                               *
0FC6  BF00 1166                ERROR  STR    $R3,<CHAN
0FC8  1CF0                            LDV    $R1,=-16             INTIALIZE LOOP COUNTER
0FC9  C866                            LDR    $R4,-$B6             DECREMENT $B6
0FCA  C866                            LDR    $R4,-$B6             DO AGAIN
0FCB  EF80 0FD2                       STB    $B6,<ERR1+4+$AF      STORE ERROR ADDRESS FOR ERROR CALL
                               ERR1   CALL   ZV$ER,$,ERMG
0FCD  FBC0 0003
0FCF  D380 0000        X
0FD1  0F80
0FD2  0FCD
0FD3  11E9
0FD4  A800 1165                       LDR    $R2,<LNBM            GET LINE NUMBER
0FD6  280D                            BLZ    $R2,>ERR6
                                      CALL   ZV$I,LINEP           PRINT "LINE"
0FD7  FBC0 0003
0FD9  D380 0000        X
0FDB  0F80
0FDC  105C
                                      CALL   ZV$TD,LNBM           PRINT THE LINE NUMBER
0FDD  FBC0 0003
0FDF  D380 0000        X
0FE1  0F80
0FE2  1165
                               *
                               * ERROR PARAMETER ROUTINE
                               *
0FE3  8751                     ERR6   CL     =$R1                 INDEX
0FE4  A810 101E                ERRP1  LDR    $R2,<LOCTAB.$R1      GET BYTE ADDRESS
0FE6  2900 0FF4                       BEZ    $R2,<PPAR            BRANCH IF DONE
0FE8  AF00 1055                       STR    $R2,<DMPLOC          OUTPUT BYTE ADDRESS
0FEA  B380 10EB                       LNJ    $B3,<SETLCT
0FEC  1055                            DC     <DMPLOC
                               *
0FED  B380 108D                       LNJ    $B3,<INBYTE          INPUT BYTE
0FEF  5048                            SOR    $R5,8                SHIFT BYTE OVER
0FF0  DF10 103A                       STR    $R5,<OUTTAB.$R1      STORE VALUE
0FF2  8AD1                            INC    =$R1
0FF3  0FF1                            B      >ERRP1               DO NEXT
                               *
                               * PRINT ERROR PARAMETERS
                               *
0FF4  9B80 103A                PPAR   LAB    $B1,<OUTTAB          GET ADDRESS OF OUTPUT TABLE
0FF6  9800 1180                       LDR    $R1,<CRLF            CR/LF
0FF8  9F00 1177                       STR    $R1,<TPR
0FFA  1CF8                            LDV    $R1,=-8              NUMB ENTRIES / LINE
                               *
                               * FIRST LINE GETS CHAN PROG INFO
                               *
0FFB  C871                     PCPV   LDR    $R4,+$B1
0FFC  4008                            SOL    $R4,8
0FFD  D871                            LDR    $R5,+$B1
0FFE  C455                            OR     $R4,=$R5             FORM VALUE
                               *
0FFF  CF00 1178                PVL    STR    $R4,<TEMP
                                      CALL   ZV$THZ,TEMP,TPR
1001  FBC0 0003
1003  D380 0000        X
1005  0F80
1006  1178
1007  1177
1008  A800 1181                       LDR    $R2,<SPACE
100A  AF00 1177                       STR    $R2,<TPR
100C  8AD1                            INC    =$R1
100D  89D1                            CMZ    =$R1
100E  0801 FFEC                       BAL    PCPV
1010  0300 1019                       BG     <GREG
1012  A800 1180                       LDR    $R2,<CRLF
1014  AF00 1177                       STR    $R2,<TPR             SET CR/LF FOR SECOND LINE
1016  9B80 205B                       LAB    $B1,<SAV5+7*$AF+1
1018  5CF7                            LDV    $R5,=-9
                               *
1019  C871                     GREG   LDR    $R4,+$B1
101A  5780 0FFF                       BINC   $R5,<PVL
101C  0F80 1057                       B      <ERR5                DONE
                               *
                               * VALUES TO INPUT
101E  0637                     LOCTAB DC     Z'0637'
101F  0737                            DC     Z'0737'
1020  2637                            DC     Z'2637'
1021  2737                            DC     Z'2737'
1022  0E37                            DC     Z'0E37'
1023  2E37                            DC     Z'2E37'
1024  3A37                            DC     Z'3A37'
1025  1437                            DC     Z'1437'
1026  1A37                            DC     Z'1A37'
1027  3E37                            DC     Z'3E37'
1028  1837                            DC     Z'1837'
1029  3837                            DC     Z'3837'
102A  1B37                            DC     Z'1B37'
102B  3B37                            DC     Z'3B37'
102C  1737                            DC     Z'1737'
102D  0337                            DC     Z'0337'
102E  3F37                            DC     Z'3F37'
102F  1D37                            DC     Z'1D37'
1030  1937                            DC     Z'1937'
1031  3C37                            DC     Z'3C37'
1032  1E37                            DC     Z'1E37'
1033  1F37                            DC     Z'1F37'
1034  0437                            DC     Z'0437'
1035  3D37                            DC     Z'3D37'
1036  0000                            RESV   4,0
                               *
                               * TABLE OF VALUES TO READ IN
                               *
103A  0000                     OUTTAB DC     0                    P MSB, RCV
103B  0000                            DC     0                    P LSB, RSV
```

```
103C  0000                       DC     0                   P MSB, XMIT
103D  0000                       DC     0                   P LSB, XMIT
103E  0000                       DC     0                   LR5 STATUS, RCV
103F  0000                       DC     0                   LR5 STATUS, XMIT
1040  0000                       DC     0                   ERROR CODE
1041  0000                       DC     0                   LR2 CONTROL WORD
1042  0000                       DC     0                   RCV LR7
1043  0000                       DC     0                   CCP FLAG
1044  0000                       DC     0                   CHAR COUNT (NEG) RCV
1045  0000                       DC     0                   CHAR COUNT (NEG) XMIT
1046  0000                       DC     0                   FRAME COUNT (NEG) , RCV
1047  0000                       DC     0                   FRAME COUNT (NEG), XMIT
1048  0000                       DC     0                   RCV CONFIG
1049  0000                       DC     0                   SIZE OF LAST XMIT BYTE
104A  0000                       DC     0                   FILL,  RCV ACTION FRAME COUNT
104B  0000                       DC     0                   RCV ACTION CHAR COUNT
104C  0000                       DC     0                   XMIT ACTION FRAME COUNT
104D  0000                       DC     0                   XMIT ACTION CHAR COUNT
104E  0000                       DC     0                   XMIT FW REV
104F  0000                       DC     0                   RCV FW REV
1050  0000                       DC     0                   SIZE OF RESIDUE EXPECTED
1051  0000                       DC     0                   UNUSED BIT MASK
1052  0000                       RESV   3,0
                          *
1055  0037                DMPLOC DC     Z'0037'             BYTE ADDRESS TO OUTPUT
1056  0000                       DC     0
1057  8FC0 0FFB           ERR5   RSTR   SAV5,=Z'FFFF'       RESTORE REGISTERS
1059  FFFF
105A  0000                       HLT                        HALT AFTER ERROR
105B  8386                       JMP    $B6                 RETURN
                          *
105C  2020 204C 494E      LINEP  TEXT   '   LINES'
105F  4524
                          *
1060  8F00 201D           DLAY   SAVE   <SAV1,=Z'CC8C'
1062  CC8C
1063  9800 1191                  LDR    $R1,<HRTZ
1065  1044                       SOR    $R1,4               SET FOR CLOCK/16
1066  0F87                       B      >DLAY2
                          *
                          * LONG DELAY (APPROX 225 MS)
                          *
1067  8F40 0FB5           DLAYLG SAVE   SAV1,=Z'CC8C'
1069  CC8C
106A  9800 1191                  LDR    $R1,<HRTZ
106C  1041                       SOR    $R1,1               GET CLOCK/4
106D  DB80 0000       X   DLAY2  LAB    $B5,<ZHPFR          CLEAR B5
106F  DF80 0001       X          STB    $B5,<ZH1SAZ+$AF     ZERO OUT LEV 1 INTERRUPT VECTOR
1071  9F00 0000       X          STR    $R1,<ZHRTCC         SET RTC CURRENT VALUE
1073  C380 0F5E                  LNJ    $B4,<UPTM           UPDATE TIME
1075  8751                       CL     =$R1
1076  9F00 0000       X          STR    $R1,<ZHRTCI         RTC RESET VALUE
1078  1C01                       LDV    $R1,=1
1079  9F00 0000       X          STR    $R1,<ZHRTCL
107B  C800 11A7                  LDR    $R4,<CONT6          INPUT STATUS FUNCTION
107D  C380 0EE8                  LNJ    $B4,<CGSCH          PUT IN CHAN NUMBER
                          *
107F  0004                       RTCN                       TURN ON RTC TIMER
1080  8980 0000       X   DLAY1  CMZ    <ZHRTCC
1082  8055                       IO     =$R5,=$R4           INPUT STAT
1083  0054
                          * HANG HERE IF MLCC DOES NOT RESPOND WITH A SECOND HALF READ
1084  DF00 115F                  STR    $R5,<STAT           STORE LAST STATUS READ
1086  0A00 1080                  BAG    <DLAY1              WAIT TILL DONE
1088  0005                       RTCF                       TURN OFF TIMER
1089  8FC0 0F93                  RSTR   SAV1,=Z'CC8C'
108B  CC8C
108C  8384                       JMP    $B4                 RETURN
                          *
                          * INPUT BYTE
                          *
108D  C800 11AB           INBYTE LDR    $R4,<CONT10         INBYTE FC
108F  C380 0EE8                  LNJ    $B4,<CGSCH          OR IN CHAN
1091  8055                ERRP2  IO     =$R5,=$R4           INPUT BYTE
1092  0054
1093  07FE                       BIOF   >ERRP2
1094  8383                       JMP    $B3
                          *
                          * BLOCK WRITE DATA TO RAM
                          *
                          *       LNJ     $B1,<SDATA
                          *       DC      DATA              LOCATION OF DATA
                          *       DC      RANGE             NUMBER OF DATA BYTES
                          *       DC      RAMAD             RAM ADDRESS
                          *       DC      EVEN              0 = EVEN BYTE CPU ADDRESS
                          *                                     BIT 15 = 1 FOR ODD BYTE START
                          *
                          * R3 MUST CONTAIN THE CHANNEL NUMBER
                          *
1095  8F00 201D           SDATA  SAVE   <SAV1,=Z'FFBF'      SAVE ALL BUT B1
1097  FFBF
1098  DCF1                       LDB    $B5,+$B1            GET ADDRESS OF DATA
1099  C871                       LDR    $R4,+$B1            GET RANGE
109A  CF00 10A7                  STR    $R4,<SPRG1          STORE RANGE
109C  DF80 10A6           SPRG4  STB    $B5,<SPRG5
109E  A871                       LDR    $R2,+$B1            GET RAM ADDRESS
109F  AF00 10A8                  STR    $R2,<SPRG2
10A1  9871                       LDR    $R1,+$B1            LOAD START BYTE INDEX
10A2  9570 7FFF                  AND    $R1,=X'7FFF'
10A4  C380 10FF                  LNJ    $B4,<MCCB           FORM CCB
10A6  1160                SPRG5  DC     <DUMMY              CPU ADDRESS
10A7  0000                SPRG1  DC     0                   RANGE
10A8  0000                SPRG2  DC     0                   RAM ADDRESS
10A9  C380 0F0B                  LNJ    $B4,<CHCT           GIVE CHANNEL CONTROL
10AB  0F82                       B      >$+2
10AC  0400                       DC     X'400'              BLOCK WRITE
                          *
                          * PROGRAM ARIVES HERE FROM SDATA OR RDATA AS WELL AS SPRG.
                          *
10AD  C380 0F1D           SPRG3  LNJ    $B4,<TEST           WAIT FOR STATUS COMPLETE, OR
10AF  0F82                       B      >$+2
10B0  000C                       DC     12                  100 MS TIMEOUT
10B1  82D5                       LB     =$R5,=X'1000'       GET STATUS COMPLETE BIT
```

```
10B2  1000
10B3  0503                     BBT     >$+2+$AF
10B4  E380 0FBA                LNJ     $B6,<ERRMB             STATUS COMPLETE NOT SET AFTER BLOCK WRITE
10B6  82D5                     LB      =$R5,=7
10B7  0007
10B8  0583                     BBF     >$+2+$AF
10B9  E380 0FBA                LNJ     $B6,<ERRMB             ERROR ; PARITY,MEMORY,OR RESOURCES
10BB  8F80 201D         SPRG7  RSTR    <SAV1,=Z'FFBF'
10BD  FFBF
10BE  8381                     JMP     $B1
                        ******
                        *
                        * BLOCK READ FROM RAM.- CHAN. NUMBER MUST BE IN R3.
                        *
                        *          LNJ       $B1,<RDATA
                        *          DC        INBUFF            INPUT BUFFER ADDRESS
                        *          DC        RANGE             NUMBER OF BYTES
                        *          DC        RAMAD             RAM ADDRESS
                        *          DC        EVEN              0 = EVEN BYTE CPU ADDRESS
                        *                                          BIT 1 = 1 FOR ODD BYTE ADDRESS
                        *                                          BIT 0 = 1 FOR NO DELAY AFTER STARTING
                        *
10BF  8F00 201D         RDATA  SAVE    <SAV1,=Z'FFBF'         SAVE EVERYTHING BUT B1.
10C1  FFBF
10C2  DCF1                     LDB     $B5,+$B1               GET IN BUFF ADD
10C3  DF80 10D0                STB     $B5,<RDTA1
10C5  C871                     LDR     $R4,+$B1               GET RANGE IN BYTES
10C6  CF00 10D1                STR     $R4,<RDTA3
10C8  9871                     LDR     $R1,+$B1
10C9  9F00 10D2                STR     $R1,<RDTA2
10CB  9871                     LDR     $R1,+$B1               PICK UP EVEN, ODD FLAG
10CC  9570 7FFF                AND     $R1,=X'7FFF'
10CE  C380 10FF                LNJ     $B4,<MCCB              FORM CCB
10D0  1160              RDTA1  DC      <DUMMY                 CPU ADDRESS
10D1  0000              RDTA3  DC      0                      RANGE
10D2  0000              RDTA2  DC      0                      RAM ADDRESS
10D3  C380 0F0B                LNJ     $B4,<CHCT              ISSUE CHANNEL CONTROL
10D5  0F82                     B       >$+2
10D6  0800                     DC      X'800'                 BLOCK READ
10D7  0F81 FFD5                B       SPRG3                  EXIT
                        *
                        *
                        * INPUT NEXT STATUS TO R5
                        *
10D9  8F00 201D         INXT   SAVE    <SAV1,=Z'0008'         B4
10DB  0008
10DC  C800 11A5                LDR     $R4,<CONT4             GET CONTROL WORD FOR INPUT NEXT STATUS
10DE  C380 0EE8                LNJ     $B4,<CGSCH             MODIFY FOR CHANNEL
10E0  8055                     IO      =$R5,=$R4              GET NEXT STATUS
10E1  0054
10E2  0703                     BIOT    >$+2+$AF
10E3  E380 0FBA                LNJ     $B6,<ERRMB             INPUT NEXT STATUS WAS NAK'ED
                        *
10E5  0F00 10D9                NOP     <INXT
10E7  8FC0 0F35                RSTR    SAV1,=Z'0008'
10E9  0008
10EA  8384                     JMP     $B4
                        *
                        * SET UP LCT FOR CHANNEL SPECIFIED IN R3
                        *
                        *              LNJ $B3,<SETLCT
                        *               -DC  TABLE
                        *
10EB  8F40 0F41         SETLCT SAVE    SAV2,=Z'E8E0'          R1,<R2,<R4,<B2,<B1
10ED  E8E0
10EE  9CF3                     LDB     $B1,+$B3               GET ADDRESS OF TABLE
10EF  8751                     CL      =$R1
                        *
10F0  A811              LCT4   LDR     $R2,$B1.$R1            GET BYTE TO OUTPUT
10F1  8AD1                     INC     =$R1
10F2  2985                     BNEZ    $R2,>LCT5              BRANCH IF NOT AT END OF TABLE
10F3  8FC0 0F39                RSTR    SAV2,=Z'E8E0'
10F5  E8E0
10F6  8383                     JMP     $B3                    RETURN
                        *
10F7  C800 11A6         LCT5   LDR     $R4,<CONT5             FUNCTION CODE FOROUT LCT BYTE
10F9  C380 0EE8                LNJ     $B4,<CGSCH             FORM IO CONTROL WORD
10FB  8052              LCT3   IO      =$R2,=$R4              OUTPUT BYTE
10FC  0054
10FD  07FE                     BIOF    >LCT3                  CHECK IF TAKEN
10FE  0FF2                     B       >LCT4                  GET NEXT BYTE
                        *
                        *
                        *         CCB FORMATION
                        *
                        *          $R3 - CONTAINS CHANNEL WANTED
                        *
                        *         LNJ      $B4,<MCCB
                        *         DC       CPU ADDRESS
                        *         DC       RANGE IN BYTES
                        *         DC       RAM ADDRESS NUMBER OR CHANNEL CONTROL WORD.
                        *
                        *
10FF  8F00 202D         MCCB   SAVE    <SAV2,=Z'FDF4'         SAVES $B1,$B3,$B2,$B5,$R7,$R5,$R4,R2,& $R1
1101  FDF4
1102  ACF4                     LDB     $B2,+$B4               LOAD $B2 WITH CPU ADDRESS
1103  D874                     LDR     $R5,+$B4               GET RANGE
1104  A874                     LDR     $R2,+$B4               PUT RAM ADDRESS IN $R2
1105  DED4                     SWB     $B5,=$B4               ALLOW $B4 TO BE USE IN SUBR. CALL
1106  C800 11A2                LDR     $R4,<CONT1             LOAD $R4 WITH I/O CONTROL WORD
1108  C380 0EE8                LNJ     $B4,<CGSCH
110A  8182                     IOLD    $B2,=$R4,=$R5          OUTPUT ADDRESS AND RANGE
110B  0054
110C  0055
110D  0703                     BIOT    >$+2+$AF
110E  E380 0FBA                LNJ     $B6,<ERRMB             ERROR, IOLD WAS NAK'ED
1110  C800 11A4                LDR     $R4,<CONT3             LOAD $R4 WITH I/O CONTROL WORD
1112  C380 0EE8                LNJ     $B4,<CGSCH             PUT I/O CONTROL WORD IN $R4
1114  8052              MCB2   IO      =$R2,=$R4              OUTPUT MLCC RAM ADDRESS
1115  0054
1116  0703                     BIOT    >$+2+$AF
1117  E380 0FBA                LNJ     $B6,<ERRMB             ERROR, OUTPUT CONTROL WAS NAK'ED
1119  CED5                     SWB     $B4,=$B5               SWAP FOR SUBR. RETURN
```

```
111A  8F80 202D           RSTR    <SAV2,=Z'FDF4'          RESTORE REGS.
111C  FDF4
111D  8384                JMP     $B4
                    *
                    *
                    *          CONVERT RTC TICKS TO MILSEC.
                    *
111E  8F00 201D     TKSEC SAVE    <SAV1,=Z'C080'          SAVE REGS.
1120  C080
1121  CF51                STR     $R4,=$R1                DUPLICATE VALUE IN $R1
1122  CB00 1193           MUL     $R4,<DIV1
1124  CF00 1186           STR     $R4,<TLOC
1126  C851                LDR     $R4,=$R1
1127  CB00 1194           MUL     $R4,<DIV2
1129  C370 000A           DIV     $R4,=10
112B  9B00 1195           MUL     $R1,<DIV3
112D  9370 0064           DIV     $R1,=100
112F  CA00 1186           ADD     $R4,<TLOC
1131  CA51                ADD     $R4,=$R1
1132  CF00 1186           STR     $R4,<TLOC
1134  8F80 201D           RSTR    <SAV1,=Z'C080'
1136  C080
1137  8385                JMP     $B5
                    *
                    *
                    *
                    *========================================================================
                    *
                    *          CONVERT MS TO BITS/SEC/10 (GIVEN CHAR. SIZE & NUMBER)
                    *
1138  8F00 201D     BPS   SAVE    <SAV1,=Z'9F80'          SAVE REGS.
113A  9F80
113B  8756                CL      =$R6                    CLEAR $R6 FOR DIVIDE
113C  B800 1168           LDR     $R3,<CHRCNT             LOAD $R3 WITH NUMBER OF CHAR.
113E  B370 000A           DIV     $R3,=10                 GET CHAR/10
1140  F870 03E8           LDR     $R7,=1000               CONVERSION FACTOR FOR MS TOC.
1142  7F08                MLV     $R7,=8                  MLVTIPLY BY CHARACTER SIZE
1143  F300 1186           DIV     $R7,<TLOC               DIVIDE BY ACTUAL TIME IN MS
1145  FF55                STR     $R7,=$R5
1146  DB53                MUL     $R5,=$R3
1147  DF00 1187           STR     $R5,<MSBS
1149  70D0                DOR     $R7,16                  SHIFT REMAINDER DOWN
114A  7F0A                MLV     $R7,=10
114B  F354                DIV     $R7,=$R4
114C  FF55                STR     $R7,=$R5
114D  DB53                MUL     $R5,=$R3
114E  D370 000A           DIV     $R5,=10
1150  DA00 1187           ADD     $R5,<MSBS
1152  70D0                DOR     $R7,16
1153  7F0A                MLV     $R7,=10
1154  F354                DIV     $R7,=$R4
1155  FB53                MUL     $R7,=$R3
1156  F370 0064           DIV     $R7,=100
1158  FA55                ADD     $R7,=$R5
1159  FF00 1187           STR     $R7,<MSBS
115B  8F80 201D           RSTR    <SAV1,=Z'9F80'
115D  9F80
115E  8385                JMP     $B5
                    ***********************************************************************
                    *
                    *
                    *
                    *
                    * CONSTANTS AND STORAGE LOCATIONS
                    *
115F  0000          STAT   DC     0                       LAST STATUS STORED HERE
1160  1160          DUMMY  DC     <DUMMY                  DUMMY ADDRESS VALUE
1161  0000          SPFLG  DC     0                       NON ZERO FOR 72 KB
1162  0000          ERRCD  DC     0                       ERROR VALUE
1163  0000          PVLU   DC     0                       CCP P COUNTER
1164  0000          FRST   DC     0                       FRST TIME FLAG
1165  0000          LNBM   DC     0                       LINE NUMBER
1166  0000          CHAN   DC     0                       CHANNEL STORED HERE
1167  0000          ELPS   DC     0                       ELAPSED TIME
1168  0000          CHRCNT DC     0                       CHARACTER COUNT
1169  0000          COUNT  DC     0                       COUNTER FOR TEST LOOPS
116A  0000          CHSZ   DC     0                       CHARACTER SIZE STORAGE
                    *                                     0->8, 1-7 -> 1-7
116B  0000          IMASK  DC     0                       MASK FOR CHANNELS PRESENT
116C  0000          RANGE  RESV   1,0
116D  0000          DEVID  RESV   1,0                     DEVICE ID
116E  FFFF          ALLONE RESV   1,-1
116F  0016          C16    DC     X'16'
1170  0000          ADSTR  RESV   1,0                     MS 6 BITS OF DEVICE ADDRESS
1171  FC00          DADD   DC     Z'FC00'                 DEFAULT DEVICE ID
1172  5555          DFLT   DC     Z'5555'                 DEFAULT VALUE
1173  0400          DRNG   DC     X'400'                  DEFAULT RANGE
1174  0000          HEAD   DC     0                       HEADER INFORMATION STORAGE
1175  0000          IFLG   DC     0                       DATA INVERSION FLAG
1176  0000          MASK   DC     0                       MASK
1177  0000          TPR    DC     0                       TEMPORARY  STORAGE
1178  0000          TEMP   DC     0                       TEMPORARY STORAGE
1179  0000          LOOP   DC     0                       LOOP COUNTER
117A  1C00          INBUF  DC     <RTB
117B  0000          EXST   DC     0                       EXPECTED STATUS FLAG
117C  0001          ERF    DC     1                       MAKE 0 FOR SHORT ERROR PRINTOUT
117D  0303          PAT    DC     X'0303'                 DATA PATTERN
117E  0000          QFLG   DC     0                       QUICK MODE FLAG
117F  0000          PFLAG  DC     0                       PRINT TEST LABLE FLAG
1180  8020          CRLF   DC     Z'8020'
1181  2020          SPACE  DC     Z'2020'
1182  0000          PASSC  DC     0                       PASS COUNTER
1183  0000          RRNG   DC     0                       RESIDUAL RANGE
1184  0000          XLOOP  DC     0                       EXTERNAL LOOP FLAG
1185  0000          CBLOOP DC     0                       CONNECTOR LOOP FLAG
1186  0000          TLOC   DC     0                       ELAPSED TIME IN MS
1187  0000          MSBS   DC     0                       BITS/SEC
1188  0000          STPTR  DC     0                       POINTER TO FRAME STATE TABLE
1189  0000          BASE   DC     0                       RAN NUMBER BASE
118A  0000          MODE   DC     0                       RAN NUMBER MODE
118B  0400          R400   DC     X'400'
118C  0085          R85    DC     X'85'
118D  0010          R10    DC     X'10'
```

```
118E  0000            C0      DC      0
118F  0008            C8      DC      8
1190  0046            C70     DC      70
1191  003C            HRTZ    DC      60                     PWR FREQ
1192  0000            SPEED   DC      0
1193  0000            DIV1    DC      0                      MSEC
1194  0000            DIV2    DC      0                      .1 MS
1195  0000            DIV3    DC      0                      .01 MS
1196  0004            C4      DC      4
1197  FFFF            CM1     DC      -1
1198  0000            TTOT    DC      0                      TOTAL OF TICKS
1199  0000            SEC     DC      0                      TOTAL SECONDS
119A  0000            MLC-FR  DC      0                      MLCC FIRMWARE REV STORED HERE
119B  0000            ODDFLG  DC      0                      NON ZERO FOR ODD RANGE
                      *
                      * HDLC FIRMWARE REV'S FOLLOW. FFFF IS DEFAULT VALUE IN THE TABLE
                      *
119C  FFFF            XHCFW1  DC      -1                     XMIT FW REV, LA 1
119D  FFFF            XHCFW2  DC      -1                     XMIT FW REV, LA 2
                      *
119E  FFFF            RHCFW1  DC      -1                     RCV FW LA 1
119F  FFFE            RHCFW2  DC      -2                     RCV FW REV LA 2

                      * BHCLA LINE SPEEDS FOLLOW.  FFFF IS THE DEFAULT VALUE
                      *
11A0  FFFF            HCLS-1  DC      -1                     SPEED/10, LA 1
11A1  FFFF            HCLS-2  DC      -1                     SPEED/10, LA 2
                      *
                      *
                      *
                      *
                      *
                      * CONTROL WORDS FOR IO OPERATIONS
                      *
                      *
                      *      I/O CONTROL WORDS
                      *
11A2  0009            CONT1   DC      Z'0009'                IOLD FUNCTION CODE
11A3  000C            CONT2   DC      Z'000C'                INPUT RANGE FUNCTION CODE
11A4  000F            CONT3   DC      Z'000F'                OUTPUT CCB CONTROL FUNCTION CODE
11A5  001A            CONT4   DC      X'1A'                  INPUT NEXT STATUS FUNCTION CODE
11A6  000B            CONT5   DC      Z'000B'                OUTPUT BYTE INTO LCT FUNCTION CODE
11A7  0018            CONT6   DC      Z'0018'                INPUT STATUS FUNCTION CODE
11A8  0005            CONT7   DC      X'5'                   OUTPUT CHANNEL CONTROL FUNTION CODE
11A9  0026            CONT8   DC      X'26'                  INPUT ID FUNCTION CODE
11AA  0001            CONT9   DC      Z'0001'                OUTPUT MLCC CONTROL FUNTION CODE
11AB  001E            CONT10  DC      X'1E'                  INPUT BYTE
11AC  0003            CONT11  DC      X'3'                   OUTPUT INTERRUPT CONTROL FUNCTION CODE
11AD  001C            CONT12  DC      X'1C'                  INPUT DATA SET STATUS
                      *
                      *

                      * STORAGE AREAS
                      *
11AE  0000            TMPSTR  RESV    10,0
                      *
11B8  0000            TSA2    DC      0
                      *
11B9  FFFF            ATLT    RESV    4,-1                   ACTIVE LINES TABLE
                      *
11BD  0000            ERAR    RESV    4,0                    ERROR BUFFER
                      * SAVE AREA -MAJOR TEST
                      *
                      *
                      * MESSAGES
                      *
11C1  4248 434C 4120  TITLE   TEXT    'BHCLA TEST'
11C4  5445 5354
                              IFZ     (SAF-2),LAF1
11C6  2044 434D 5333          TEXT    ' DCMS3, SAF-A '
11C9  2C20 5341 462D
      4120
11CD  2020 4A55 4C20  DATE    TEXT    '  JUL 24, 1978$'
11D0  3234 2C20 3139
      3738 2400
11D5  4248 434C 4120  MESG1   TEXT    'BHCLA CHANNEL(S)$'
11D8  4348 414E 4E45
      4C28 5329 2400
11DE  4E45 5854 2400  MESG2   TEXT    'NEXT$'
11E1  5057 5220 4652  MESG5   TEXT    'PWR FREQ (HZ) $'
11E4  4551 2028 485A
      2920 2400
11E9  7878 2020 2400  ERMG    TEXT    'XX  $'
11EC  2054 5241 4E53  XPRT    TEXT    ' TRANSMIT = $'
11EF  4D49 5420 3D20
      2400
11F3  2052 4356 203D  RPRT    TEXT    ' RCV = $'
11F6  2024
11F7  204C 494E 4520  IDMSG   TEXT    ' LINE $'
11FA  2400
11FB  2049 4420 3D20  EQ      TEXT    ' ID = $'
11FE  2400
                      *
                      * -VC-VH-VA-VN-VN-VE-VL PROGRAMS GO HERE
                      *
                      *
                      *
                      *
                      *
                      *
                      * SOURCE = DCMS3A4-CCP
                      *
                      * MAR 24, 1978
                      *
                      *
                      * BROADBAND CHANNEL FUNCTIONAL PROGRAM, RECEIVE
                      *
                      *       ORG     X'200'
      11FF            CCP1    EQU     $
                      *       B       RCV1
11FF  E007
                      *       LOC     LAST
      0202            LAST    EQU     X'0202'
```

```
004511                       *
004512                       *       WAIT    END                 OF FRAMES
004513                       *       LD      =11
004514    1200  0190
004517                       *       JUMP    EREND               CRI AFTTER LAST MESSAGE
004518    1201  0BE6
004521    1202  0195                 DC      EREND-X'0207'
004522                       *
004523                       *------------------
004524                       *
004525                       * RCV INITIALIZATION, RESYNC
004526                       *
004527                       *
004528                       *       LOC     RCV1
004529          0208         RCV1    EQU     X'0208'
004530                       *       LD      X'3E'               GET CONTROL FLAG
004533    1203  503E
004534                       *       AND     =X'8'               STRIP TO READ CRC BIT
004537    1204  9308
004538                       *       OR      =X'11'              INITIALIZATION PARAMETER
004541    1205  9411
004542                       *       OUT     6
004545                       *
004546                       *       LD      =6                  TEST, RCV ON
004547    1206  3690
004550                       *       OUT     2
004553    1207  0632
004554                       *
004555                       *
004556                       *
004557                       *       LOC     RCV2
004558          0212         RCV2    EQU     X'0212'
004559                       *       WAIT
004560                       *-------------------------------------------------------------------
004561                       *
004562                       * READ FW REV
004563                       *
004564                       *
004565                       *       IN      5                   INPUT DATA SET STATUS
004568    1208  0125
004569                       *       ST      X'E'
004572    1209  510E
004573                       *       RECV    0                   INPUT FW REV + ENABLE RECV
004576                       *       ST      X'1F'
004577    120A  A051
004580                       *
004581                       *       LOC     RCV3
004582          0219         RCV3    EQU     X'0219'
004583                       *       WAIT
004584    120B  1F01
004585                       *-------------------------------------------------------------------
004586                       *
004587                       * PROCESS THE INITIAL ILS RUPT
004588                       *
004589                       *       IN      7                   INPUT RCV STATUS
004592                       *       ST      X'1A'
004593    120C  2751
004596                       *       C       =X'80'              SHOULD HAVE ILS RUPT
004597    120D  1A92
004600                       *       BET     RCV4
004601    120E  80E1
004604                       *       LD      =1
004605    120F  0690
004608                       *       JUMP    EREND
004609    1210  01E6
004612    1211  0177                 DC      EREND-X'0225'
004613                       *
004614                       *
004615                       *       LOC     RCV4
004616          0226         RCV4    EQU     X'0226'
004617                       *       IN      7                   READ LR7 AGAIN
004620                       *       ST      X'1A'
004621    1212  2751
004624                       *       C       =X'80'
004625    1213  1A92
004628                       *       BET     RCV4A
004629    1214  80E1
004632                       *
004633                       *       LD      =13
004634    1215  0690
004637                       *       JUMP    EREND
004638    1216  0DE6
004641    1217  016B                 DC      EREND-X'0231'
004642                       *
004643                       *       LOC     RCV4A
004644          0232         RCV4A   EQU     X'0232'
004645                       *       IN      5
004648                       *       ST      X'E'
004649    1218  2551
004652                       *       AND     =8
004653    1219  0E93
004656                       *       BZF     RCV5                BRANCH IF ADAPT RDY = 1
004657    121A  08F2
004660                       *       LD      =2                  ERROR CODE
004661    121B  0690
004664                       *       JUMP    EREND
004665    121C  02E6
004668    121D  015F                 DC      EREND-X'023D'
004669                       *
004670                       *       LOC     RCV5
004671          023E         RCV5    EQU     X'023E'
004672                       *       BART    RCV6
004675    121E  E506
004676                       *       LD      =3
004679    121F  9003
004680                       *       JUMP    EREND
004683    1220  E601
004684                       *
004685                       *       LOC     RCV6
004686          0245         RCV6    EQU     X'0245'
004687                       *       RECV    0                   DUMMY TO ADVANCE
004690    1221  58A0
004691                       *       BARF    RCV7
```

```
1222   F506
                       *        LD      =9
1223   9009
                       *        JUMP    EREND              ADAP RDY DIDN'T RESET
1224   E601
                       *
                       *        LOC     RCV7
       024D            RCV7     EQU     X'024D'
                       *        IN      5                  IN DATA SET STAT
1225   5025
                       *        ST      X'E'
1226   510E
                       *        AND     =X'4'              STRIP TO READY
1227   9304
                       *        BZT     RCV8
1228   E206
                       *
                       *        LD      =4
1229   9004
                       *        JUMP    EREND
122A   E601
                       *
                       * OUTPUT RCV CONFIGURATION
                       *
                       *        LOC     RCV8
       0259            RCV8     EQU     X'0259'
                       *        LD      X'17'
122B   4450
                       *
                       * IF TCB IS SET MODIFY BYTE SIZE SPECIFIED
                       * BY  LR6 (SHOULD BE DON'T CARE)
                       *
                       *        AND     =X'20'             TCB BIT
122C   1793
                       *        BZT     DOCFG              B = NO TCB
122D   20E2
                       *        LD      X'17'              GETVALUE
122E   0750
                       *        XOR     =4     .           CHANGE BIT 5
122F   1795
                       *        B       OUTIT
1230   04E0
                       *
                       *        LOC     DOCFG
       0265            DOCFG    EQU     X'0265'
                       *        LD      X'17'
1231   0350
                       *
                       *        LOC     OUTIT
       0267            OUTIT    EQU     X'0267'
                       *        OUT     6
1232   1736
                       *
                       * TURN ON XMIT, RCV, OR BOTH
                       *
                       *        LD      X'14'              PICK UP CONTROL
1233   5014
                       *        OUT     2
                       *
                       *        LOC     DLOOP
       026B            DLOOP    EQU     X'026B'
                       *
                       *        LD      =X'FF'
1234   3290
                       *        ST      X'1B'              SET INITIAL FRAME COUNT
1235   FF51
                       *        LOC     XYZ
       026F            XYZ      EQU     X'026F'
                       *        WAIT
1236   1B01
                       *
                       *---------------------------------------------------------------
                       *
                       *   DECIDE TO DO LONG OR SHORT DATA LOOP
                       *
                       *
                       *
                       *        LOC     LP1
       0270            LP1      EQU     X'0270'
                       *        LD      X'3E'              GET FLAG
1237   503E
                       *        AND     =X'90'             STRIP TO OVERUN BIT
1238   9390
                       *        BZF     LUP2               BRANCH IF LONG LOOP
1239   F204
                       *        JUMP    RCV10              DO SHORT LOOP
123A   E600
                       *
                       * LONG DATA LOOP - 1ST SEE IF TIME FOR ANY SPECIAL ACTION
                       *
                       *        LOC     LUP2
       0279            LUP2     EQU     X'0279'
                       *        LD      X'3F'              GET ACTION FRAME
123B   2C50
                       *        OR      =X'C0'             MAKE 8 BITS
123C   3F94
                       *        C       X'1B'              COMPARE WITH ACTUAL
123D   C052
                       *        BEF     RCV9               B = NO OVRUN NOW
123E   1BF1
                       *-------------
                       *
                       * SPECIAL ACTION
                       *
                       *        LD      X'1D'              GET ACTION CHAR
123F   0C50
                       *        C       X'18'              COMPARE WITH ACTUAL
1240   1D52
                       *        BEF     RCV9               B = NO OVERUN NOW
1241   18F1
                       *        LD      =X'20'
1242   0690
                       *        JUMP    RDLAY              DELAY ABOUT 100 MS
```

```
1243  20E6
1244  0090            DC      RDLAY-X'028B'
                *-------------
                *
                * NO SPECIAL ACTION
                *
                *
                *       LOC     RCV9
      028C      RCV9    EQU     X'028C'
                *       IN      7                   GET STATUS
                *       ST      X'1A'
1245  2751
                *       BZF     RC5                 BRANCH IF STATUS TO READ
1246  1AF2
                *       RECV    0
1247  22A0
                *       ST
                *       LD      X'18'
1248  1150
                *       DEC
1249  1805
                *       ST      X'18'               UPDATE CHAR COUNT
124A  5118
                *       BLCF    NEXT
124B  F306
                *       LOC     ER-5
      029A      ER-5    EQU     X'029A'
                *       LD      =5                  MISSED CLOSED FLAG
124C  9005
                *       JUMP    EREND
124D  E600
                *       LOC     NEXT
      029F      NEXT    EQU     X'029F'
                *       BART    LUP2
124E  FEE5
                *       WAIT
124F  D901
                *       B       LUP2
1250  E0D6
                *----------------------------------------------------------------------
                *
                * MINIMUM LOOP
                *
                *       LOC     RCV10
      02A4      RCV10   EQU     X'02A4'
                *       IN      7
                *       ST      X'1A'
1251  2751
                *       BZF     RC5                 B = SOME STATUS TO READ
1252  1AF2
                *       RECV    0
1253  0AA0
                *       ST
                *       BLCT    ER-5                BR = MISSED CLOSE FLAG
1254  11E3
                *       BART    RCV10
1255  EEE5
                *       WAIT
1256  F601
                *       B       RCV10
1257  E0F3
                *
                *
                *----------------------------------------------------------------------
                *
                * COME HERE WHEN NON -ZERO LR7 FOUND IN DATA LOOP
                *
                *       LOC     RC5
      02B2      RC5     EQU     X'02B2'
                *       AND     =X'F0'              STRIP OFF RESIDUE
1258  93F0
                *       C       =X'20'              CHECK FOR DIAGNOSTIC STATUS
1259  9220
                *       BEF     NOTEST              BR IF NOT DIAGNOSTIC CRC MODE
125A  F10C
                *       LD      X'3E'               GET FLAG
125B  503E
                *       AND     =X'8'               READ CRC BIT
125C  9308
                *       BZF     NOTESM
125D  F20A
                *       LD      =X'10'
125E  9010
                *       JUMP    EREND               ERROR, DIAGNOSTIC STATUS WHEN NOT
125F  E600
                *                                 IN DIAGNOSTIC MODE
                *       LOC     NOTEST
      02C3      NOTEST  EQU     X'02C3'
                *-------------
                *
                * CHECK FOR NORMAL CLOSING STATUS, BITS 0 - 4
                *
                *
                *
                *       C       =X'10'
1260  DA92
                *       BEF     JMP2                BRANCH IF ERROR
1261  10F1
                *       LOC     NOTESM
      02C7      NOTESM  EQU     X'02C7'
                *       LD      =X'10'
1262  5290
                *       OR      X'11'
1263  1054
                *       ST      X'11'               SET CLOSE BIT
1264  1151
                *-------------
                *
                * CHECK THAT LAST BYTE SIZE IS AS EXPECTED
                *
                *
                *       LD      X'1A'               GET STATUS
1265  1150
```

```
                    *         AND      =X'F'               STRIP TO RESIDUE BITS
1266   1A93
                    *                                    ....AND PARTIAL BYTE BIT
                    *         C        4                   COMPARE WITH WHAT SHOULD BE
1267   0F52
                    *         BET      RCV12
1268   04E1
                    *         LD       =10
1269   0690
                    *         JUMP     EREND               RESIDUE WRONG FOR LAST BYTE
126A   0AE6
126B   00C3                  DC       EREND-X'02D9'
                    *                                    ....OR PARTIAL BYTE BIT WRONG
                    *         LOC      RCV12
       02DA         RCV12     EQU      X'02DA'
                    *         RECV     0
                    *         ST
126C   A011
                    *         LD       X'18'
126D   5018
                    *         DEC
                    *         ST       X'18'               DEC CHAR COUNT
126E   0551
                    *         LD       =X'10'
126F   1890
                    *         OR       X'11'
1270   1054
                    *         ST       X'11'               SET CLOSE BIT
1271   1151
                    *         LD       X'3E'               GET FLAG
1272   1150
                    *         AND      =X'8'               STRIP TO READ CRC DIAGNOSTIC BIT
1273   3E93
                    *         BZT      RCV11
1274   08E2
                    *         WAIT
1275   0E01
                    *------------------------------------------------------------------------
                    *
                    * RECEIVE CRC RESIDUES IF IN CRC DIAGNOSTIC MODE
                    *
                    *         RECV     0                   GETS RES BYTE 1
                    *         ST
1276   A011
                    *         WAIT
                    *         RECV     0                   GETS RES BYTE 2
1277   01A0
                    *         ST
                    *         WAIT
1278   1101
                    *         RECV     0                   GETS XMIT CRC BYTE 1
                    *         ST
1279   A011
                    *         WAIT
                    *         RECV     0                   GETS XMIT CRC BYTE 2
127A   01A0
                    *         ST
                    *         WAIT
127B   1101
                    *         LOC      RCV11
       02FA         RCV11     EQU      X'02FA'
                    *         GNB
                    *
                    *-------------
                    *
                    * END OF FRAME PROCESSING
                    *
                    *         LOC      EOFP
       02FB         EOFP      EQU      X'02FB'
                    *         LD       X'1B'               GET FRAME COUNT
127C   0250
                    *         DEC
127D   1B05
                    *         ST       X'1B'
127E   511B
                    *         BLBF     RC92                BRANCH IF MORE DATA COMING
127F   F434
                    *
                    *-------------
                    *
                    * LAST FRAME
                    *         LOC      LFRM
       0302         LFRM      EQU      X'0302'
                    *
                    *         LD       X'3F'               GET FILL MODE
1280   503F
                    *         AND      =X'80'
1281   9380
                    *         BZF      RC91                BRANCH IF FLAG IDLE
1282   F22B
                    *         BART     EOFR
1283   E502
                    *         WAIT
                    *         LOC      EOFR
       030B         EOFR      EQU      X'030B'
                    *         IN       7                   GET STATUS
1284   0127
                    *         ST       X'1A'
1285   511A
                    *         C        =X'80'
1286   9280
                    *         RECV     0                   DUMMY RECV FOR ILS
                    *         BET      RC91                DONE WITH FRAMES
1287   A0E1
                    *         LOC      RC90
       0313         RC90      EQU      X'0313'
                    *         LD       =6
1288   2090
                    *         JUMP     EREND               SHOULD HAVE ILS SET
1289   06E6
128A   0085                  DC       EREND-X'0317'
                    *
                    *         LOC      JMP2
```

```
        0318   JMP2    EQU     X'0318'
               *       JUMP    STATER
128B    E600
               *
               *-------------
               *
               * DELAY ROUTINE USED TO INDUCE OVERRUN
               *
               * RDLAY
               *       LOC     RDLAY
        031B   RDLAY   EQU     X'031B'
               *       ST      X'1C'                STORE COUNT
128C    4551
               * RCCPD1
               *       LOC     RCCPD1
        031D   RCCPD1  EQU     X'031D'
               *       LD      =0
128D    1C90
               *RCCPD2
               *       LOC     RCCPD2
        031F   RCCPD2  EQU     X'031F'
               *       DEC
128E    0005
               *       AND     =X'FF'
128F    93FF
               *       AND     =X'FF'
1290    93FF
               *       AND     =X'FF'
1291    93FF
               *       BZF     RCCPD2
1292    F2F8
               *       LD      X'1C'
1293    501C
               *       DEC
               *       ST      X'1C'
1294    0551
               *       BZF     RCCPD1
1295    1CF2
               *       JUMP    RCV9                 END OF DELAY ROUTINE
1296    EFE6
1297    FF5B           DC      RCV9-X'0331'
               *-------------
               *
               * LAST FRAME, FLAG IDLE
               *
               *       LOC     RC91
        0332   RC91    EQU     X'0332'
               *
               *       JUMP    LAST
1298    E6FE
               *------------------------------------------------------------------
               *
               * INTERFRAME STATE
               *
               *
               *       LOC     RC92
        0335   RC92    EQU     X'0335'
               *       BART    RCNXT
1299    CEE5
               *       WAIT
129A    0201
               *       LOC     RCNXT
        0338   RCNXT   EQU     X'0338'
               *       IN      7                    GET RCV STATUS
               *       ST      X'1A'
129B    2751
               *       LD      X'3F'                XMIT CONFIG.
129C    1A50
               *       AND     =X'80'
129D    3F93
               *       BZF     RC89                 BRANCH IF FLAG IDLE
129E    80F2
               *-------------
               *
               * ABORT IDLE STATE. CHECK FOR ILS STATUS RECEIVED.
               * ILS STATUS RECEIVED.
               *
               *
               *       LD      X'1A'
129F    1850
               *       NOP
12A0    1A00
               *       C       =X'80'
12A1    9280
               *       BET     RC88                 BRANCH IF  ILS
12A2    E106
               *
               *       LD      =7                   SHOULD BE ILS STATUS, IS NOT
12A3    9007
               *       JUMP    EREND
12A4    E600
               *       LOC     RC88
        034D   RC88    EQU     X'034D'
               *
               *       RECV    0                    DUMMY RECV
12A5    50A0
               *       WAIT
               *------------------------------------------------------------------
               * CRI TO GET HERE IS 1ST DATA FOR NEXT FRAME
               *
               *       BART    RC89
12A6    01E5
               *       LD      =8
12A7    0890
               *       ST      X'3A'
12A8    0851
               *       JUMP    EREND                ADAP READY NOT SET AFTER DATA CRI
12A9    3AE6
12AA    0045           DC      EREND-X'0357'
               *
               * FIRST DATA, NEXT FRAME
               *
               *       LOC     RC89
               0358     RC89   EQU    X'0358'
                        *
                        *      LD     =0
        12AB   9000
                        *      ST     X'18'          CLEAR CHAR COUNT
        12AC   5118
                        *      JUMP   XYZ
        12AD   E6FF
                        *
                        *-------------------------------------------------------------------
                        *
                        * COME HERE FOR STATUS OTHER THAN CLOSE
                        *
                        *      LOC    STATER
               035F     STATER EQU    X'035F'
                        *      LD     X'1A'          GET STATUS
        12AE   1150
                        *      AND    =X'10'         STRIP TO EOF
        12AF   1A93
                        *      BZT    STAT1
        12B0   10E2
                        *      LD     =X'10'
        12B1   0790
                        *      OR     X'11'
        12B2   1054
                        *      ST     X'11'
        12B3   1151
                        *      LOC    STAT1
               036B     STAT1  EQU    X'036B'
                        *      LD     X'1A'
        12B4   1150
                        *      AND    =X'80'         STRIP TO ABORT/IDLE
        12B5   1A93
                        *      BZT    STAT2
        12B6   80E2
                        *      LD     =X'2'
        12B7   0790
                        *      OR     X'10'
        12B8   0254
                        *      ST     X'10'
        12B9   1051
                        *      LOC    STAT2
               0377     STAT2  EQU    X'0377'
                        *      LD     X'1A'
        12BA   1050
                        *      AND    =X'40'         STRIP TO OVER RUN
        12BB   1A93
                        *      BZT    STAT3
        12BC   40E2
                        *      LD     =X'20'
        12BD   0790
                        *      OR     X'10'
        12BE   2054
                        *      ST     X'10'
        12BF   1051
                        *      LOC    STAT3
               0383     STAT3  EQU    X'0383'
                        *      LD     X'1A'
        12C0   1050
                        *      AND    =X'20'         STRIP TO FCS ERROR
        12C1   1A93
                        *      BZT    STAT4
        12C2   20E2
                        *      LD     =X'40'
        12C3   0790
                        *      OR     X'11'
        12C4   4054
                        *      ST     X'11'
        12C5   1151
                        *      LOC    STAT4
               038F     STAT4  EQU    X'038F'
                        *      RECV   0              DUMMY TO ADVANCE
        12C6   11A0
                        *      BLCT   STAT5          CHECK BUFFER NOT FULL
        12C7   E302
                        *      ST
                        *      LOC    STAT5
               0393     STAT5  EQU    X'0393'
                        *      GNB
        12C8   1102
                        *      BLBF   RC93           BRANCH IF MORE COMING
        12C9   F404
                        *      JUMP   LFRM
        12CA   E6FF
                        *      LOC    RC93
               0399     RC93   EQU    X'0399'
                        *      JUMP   EOFP
        12CB   6AE6
        12CC   FF60            DC     EOFP-X'039B'
                        *
                        *-------------------------------------------------------------------
                        *
                        * FATAL ERROR ROUTINE
                        *
                        *      LOC    EREND
               039C     EREND  EQU    X'039C'
                        *      ST     X'3A'
        12CD   513A
                        *      LD     =0
        12CE   9000
                        *      OUT    2              SHUT EVERYTHING OFF
                        *      NOP
        12CF   3200
                        *      NOP
                        *      NOP
        12D0   0000
                        *      NOP
                        *      LOC    LUP
               03A5     LUP    EQU    X'03A5'
                        *      WAIT
        12D1   0001
                        *      B      LUP
        12D2   E0FE
```

```
                    *       NOP
                    *       NOP
12D3  0000
                    *------------------------------------------------------------
                    *------------------------------------------------------------
                    *------------------------------------------------------------
                    *
                    *
                    * BROADBAND CHANNEL FUNCTIONAL PROGRAM , XMIT
                    *
                    *       ORG     X'400'
      12D4          CCP2    EQU     $
                    *       B       XMIT21
12D4  E007
                    *
                    *       LOC     LAST1
      0402          LAST1   EQU     X'0402'
                    *       WAIT
                    *       LD      =12
12D5  0190
                    *       JUMP    EREND             CRI AFTER XMIT TURNED OFF
12D6  0CE6
12D7  FF95                  DC      EREND-X'0407'
                    *
                    *-------------
                    *
                    * DECIDE TO READ FW REV OR NOT
                    *
                    *
                    *       LOC     XMIT21
      0408          XMIT21  EQU     X'0408'
                    *       LD      X'39'             GET READ REV FLAG
12D8  5039
                    *       BZF     RREV              B = READ FW REV
12D9  F20F
                    *
                    *-------------
                    *
                    * DON'T READ FW REV.
                    *
                    *
                    *
                    *       LD      =X'45'
12DA  9045
                    *       OUT     2                 TON, RQS OFF
                    *       LD      =X'18'
12DB  3290
                    *       OUT     7                 TEOF OFF
12DC  1837
                    *       NOP
                    *       NOP
12DD  0000
                    *       LD      =0
12DE  9000
                    *       SEND    0                 DUMMY OUTPUT
                    *       WAIT
12DF  6001
                    *       B       XREV
12E0  E01B
                    *-------------
                    *
                    * READ XMIT FW REV NUMBER
                    *
                    *
                    *       LOC     RREV
      041A          RREV    EQU     X'041A'
                    *       LD      =X'45'
12E1  9045
                    *       OUT     2                 TEST, XMIT
                    *       LD      =X'19'            EOF BIT
12E2  3290
                    *       OUT     7
12E3  1937
                    *       LD      =0
12E4  9000
                    *
                    *       SEND    0                 DUMMY OUTPUT
                    *       NOP
12E5  6000
                    *       WAIT
                    *
                    *       LOC     XREVX
      0425          XREVX   EQU     X'0425'
                    *       IN      5                 READ STAT
12E6  0125
                    *       ST      X'2E'
12E7  512E
                    *       AND     =X'1'             STRIP TO UND RUN BIT
12E8  9301
                    *       BZT     XREV              IF ZERO DONE
12E9  E209
                    *       LD      X'1E'             GET FIRMWARE REV
12EA  501E
                    *       DEC
                    *       ST      X'1E'
12EB  0551
                    *       WAIT
12EC  1E01
                    *       B       XREVX
12ED  E0F2
                    *-------------
                    *
                    * SET INITIAL FILL STATE
                    *
                    *
                    *       LOC     XREV
      0434          XREV    EQU     X'0434'
                    *       LD      =4
12EE  9004
                    *       OUT     2                 SHUT OFF XMIT
                    *
                    *       LD      X'3F'             GET IDLE STATE
```

```
12EF   3250
                        *       AND     =X'20'              STRIP TO PERTINENT BIT
12F0   3F93
                        *       BZT     XSTAR               BRANCH IF ABORT IDLE
12F1   20E2
                        *       LD      =X'10'              SET FOR FLAG IDLE
12F2   0390
                        *       LOC     XSTAR
       043F             XSTAR   EQU    X'043F'
                        *       OUT     6                   SET FILL MODE
12F3   1036
                        *       WAIT
                        *------------------------------------------------------------------------
                        *
                        * GET HERE BY CRI AFTER RCV SIDE TURNS ON XMIT
                        *
                        *       LD      =X'1C'
12F4   0190
                        *       OUT     7                   TEOF, 0 FILLS
12F5   1C37
                        *       LD      =0
12F6   9000
                        *       SEND    0                    DUMMY SEND
                        *       WAIT
12F7   6001
                        *       NOP
                        *       NOP
12F8   0000
                        *       NOP
                        *       NOP
12F9   0000
                        *       NOP
                        *       NOP
12FA   0000
                        *------------------------------------------------------------------------
                        *
                        * START OF FRAME INITIALIZATION
                        *
                        *       LOC     SFRM
       044E             SFRM    EQU    X'044E'
                        *       WAIT
                        *
                        *       LD      X'37'
12FB   0150
                        *       AND     =X'10'              STRIP TO EOF BIT
12FC   3793
                        *       BZT     SFRM3               B = NO EOF BIT SET
12FD   10E2
                        *
                        * INITIAL EOF BIT IS SET
                        *
                        *
                        *       LD      X'37'               GET CONFIG WORD AGAIN
12FE   0750
                        *       AND     =X'E'               STRIP TO BYTE SIZE INFO
12FF   3793
                        *       B       SFRM2
1300   0EE0
                        *
                        *
                        * INITIAL EOF BIT NOT SET
                        *
                        *       LOC     SFRM3
       045B             SFRM3   EQU    X'045B'
                        *       LD      X'37'
1301   0F50
                        *
                        * IF TCB SPECIFIED GIVE SCREWED UP BYTE SIZE TO LR7.
                        * TCB SHOULD OVERIDE THIS
                        *
                        *       AND     =X'20'              STRIP TO TCB BIT
1302   3793
                        *       BZT     SFRM1
1303   20E2
                        *       LD      X'37'               GET VALUE FOR LR7
1304   0750
                        *       XOR     =4                  INVERT BIT 5
1305   3795
                        *       B       SFRM2
1306   04E0
                        *
                        *       LOC     SFRM1
       0467             SFRM1   EQU    X'0467'
                        *       LD      X'37'
1307   0350
                        *
                        *-------------
                        *
                        * SET UP XMIT CONFIGURATION
                        *
                        *       LOC     SFRM2
       0469             SFRM2   EQU    X'0469'
                        *       OUT     7                   TRANSMIT CONFIGURATION
1308   3737
                        *       LD      X'3F'               GET FILL  MODE AND DELAY XMIT BIT
1309   503F
                        *       AND     =X'80'
130A   9380
                        *       SR
                        *       SR
130B   0707
                        *       OR      X'3F'
130C   543F
                        *       AND     =X'60'
130D   9360
                        *       SR
                        *       OUT     6                   SET INTER-FILL MODE
130E   0736
                        *       LD      X'3B'
130F   503B
                        *       DEC
                        *       ST      X'3B'               DEC FRAME COUNT
1310   0551
```

```
                          *
                          * SHOULD WE TAKE LONG OR SHORT DATA LOOP?
                          *
                          *       LOC    TDATA
        047B              TDATA  EQU    X'047B'
                          *
                          *       LD     X'3E'              GET FLAG
 1311   3B50
                          *       AND    =X'83'
 1312   3E93
                          *       BZT    TDAT9              BR = DO MINIMUM LOOP
 1313   83E2
                          *-------------
                          *
                          * LONG DATA LOOP, 1ST SEE IF ANY SPECIAL ACTION TO BE TAKEN.
                          *
                          *       LOC    TXYZ
        0481              TXYZ   EQU    X'0481'
                          *
                          *       LD     X'19'              ACTION FRAME
 1314   3E50
                          *       C      X'3B'              COMPARE WITH ACTUAL
 1315   1952
                          *       BEF    TDAT3              B = NOT THIS FRAME
 1316   3BF1
                          *
                          *       LD     X'3C'              GET CHAR
 1317   2250
                          *       C      X'38'
 1318   3C52
                          *       BEF    TDAT3              B = NOT THIS CHAR
 1319   38F1
                          *
                          * IT IS THE CHAR AND FRAME TO TAKE SPECIAL ACTION.
                          *
                          *       LD     X'3E'
 131A   1C50
                          *       AND    =X'2'              STRIP TO ABORT
 131B   3E93
                          *       BZT    TDAT2              B = NO ABORT
 131C   02E2
                          *       LD     =X'90'             ABORT
 131D   0790
                          *       OUT    7                  OUT CONFIG
 131E   9037
                          *       LD
                          *       B      TDAT10             GO TO END OF FRAME PROCESSING
 131F   10E0
                          *
                          *       LOC    TDAT2
        0499              TDAT2  EQU    X'0499'
                          *       LD     X'3E'              GET FLAG
 1320   5250
                          *       AND    =1                 STRIP TO DELAY FLAG
 1321   3E93
                          *       BZT    TDAT5              DON'T DELAY HERE
 1322   01E2
                          *       LD     =20
 1323   0690
                          *       JUMP   XDLAY              DELAY 100 MS
 1324   14E6
 1325   00DA                     DC     XDLAY-X'04A3'
                          *       LOC    TDAT5
        04A4              TDAT5  EQU    X'04A4'
                          *       NOP
                          *       NOP
 1326   0000
                          *       NOP
                          *       NOP                       PATCH SPECIAL ACTION HERE
 1327   0000
                          *-------------
                          *
                          * NO SPECIAL ACTION
                          *
                          *       LOC    TDAT3
        04A8              TDAT3  EQU    X'04A8'
                          *       LD     X'38'
 1328   5038
                          *       DEC
                          *       ST     X'38'              DEC CHAR COUNT
 1329   0551
                          *       AND    =16
 132A   3893
                          *       C      =0
 132B   1092
                          *       LD
 132C   0010
                          *       BEF    TDAT4
 132D   F103
                          *       OR     X'3D'              PUT IN LEADING SPACES 1/3 OF TIME
 132E   543D
                          *       LOC    TDAT4
        04B6              TDAT4  EQU    X'04B6'
                          *       BLCT   EOF
 132F   E310
                          *       SEND   0
                          *       BART   TXYZ               MORE SPACE IN BUFFER
 1330   60E5
                          *       WAIT
 1331   C701
                          *       B      TXYZ
 1332   E0C4
                          *----------------------------------------------------------------
                          *
                          * MINIMUM LOOP
                          *
                          *       LOC    TDAT9
        04BE              TDAT9  EQU    X'04BE'
                          *       LD
                          *       BLCT   EOF
 1333   10E3
                          *       SEND   0
 1334   0760
```

```
                      *       BART    TDAT9
        1335  E5FB
                      *       WAIT
                      *       B       TDAT9
        1336  01E0
                      *-----------------------------------------------------------------
                      *
                      * END OF FRAME ROUTINE
                      *
                      *
                      *       LOC     EOF
              04C7    EOF     EQU     X'04C7'
                      *       ST      X'39'               TEMP STORAGE
        1337  F851
                      *       LD      3                   GET LAST BYTE INFO
        1338  3950
                      *       AND     =X'F'               STRIP TO BYTE SIZE AND PARTIAL BYTE
        1339  0393
                      *       ST      X'24'
        133A  0F51
                      *       LD      X'19'               GET CONTROL FRAME NUMBER
        133B  2450
                      *       C       X'3B'               CHECK WITH ACTUAL
        133C  1952
                      *       BET     EOF1                B = IS CONTROL FRAME
        133D  3BE1
                      *
                      * IF IT IS A CONTROL FRAME PICK UP LAST CONTROL WORD
                      * FROM X'37'. OTHERWISE SEND ONLY EOF +BYTE SIZE.
                      *
                      *       NOP
        133E  0B00
                      *       NOP
                      *       NOP
        133F  0000
                      *       NOP
                      *       LD      X'24'               GET BYTE SIZE
        1340  0050
                      *       OR      =X'10'              SET EOF BIT
        1341  2494
                      *       B       EOF2
        1342  10E0
                      *
                      *       LOC     EOF1
              04DF    EOF1    EQU     X'04DF'
                      *       LD      X'37'               GET XMIT CONFIG INFO
        1343  0950
                      *       OR      =X'10'              PUT IN EOF
        1344  3794
                      *       AND     =X'30'              STRIP OFF BYTE SIZE
        1345  1093
                      *-------------
                      *
                      * SEND OUT CLOSE FLAG
                      *
                      *       OR      X'24'               PUT IN BYTE SIZE FOR LAST
        1346  3054
                      *       LOC     EOF2
              04E7    EOF2    EQU     X'04E7'
                      *       OUT     7                   TELL HDLC TO SEND CLOSE FLAG
        1347  2437
                      *       LD      X'39'
        1348  5039
                      *       LOC     TDAT10
              04EA    TDAT10 EQU      X'04EA'
                      *       SEND    0
                      *       LD      X'38'
        1349  6050
                      *       DEC
        134A  3805
                      *       ST      X'38'               DEC CHAR COUNT
        134B  5138
                      *
                      * TEST TO SEE IF "DELAY" XMIT BIT IS ON.
                      *
                      *       LD      X'3F'
        134C  503F
                      *       AND     =X'40'              GET DELAY XMIT BIT
        134D  9340
                      *       BZT     TDAT12              B = NOT SET
        134E  E218
                      * GIVE DELAY FOR TIME FOR XMIT TO OCCUR ID "DELAY XMIT" BROKEN
                      *       JUMP    XDLAY1
        134F  E600
                      *
                      * CHECK RCV FRAME + CHAR COUNT TO INSURE NO CHAR HAVE BEEN
                      * RECEIVED WHILE "DELAY XMIT" WAS ON.
                      *
                      *       LOC     TDAT13
              04F9    TDAT13 EQU      X'04F9'
                      *       LD      X'1B'               GET RCV FRAME COUNT
        1350  8A50
                      *       C       =X'FF'
        1351  1B92
                      *       BEF     TONER               B = ERROR, FRAME RCV'D WITH
        1352  FFF1
                      *                                 XMIT DELAY BIT ON
                      *       LD      X'18'               GET RCV CHAR COUNT
        1353  3F50
                      *       BZF     TONER               B = ERROR, CHAR RCV WITH
        1354  18F2
                      *                                 XMIT DELAY BIT ON
                      *       LD      X'3F'               GET CONFIGURATION WORD
        1355  3B50
                      *       AND     =X'80'              XMIT LR6 CONFIGURATION
        1356  3F93
                      *       ST      X'3F'
        1357  8051
                      *       SR
        1358  3F07
                      *       SR
                      *       SR
        1359  0707
```

```
                   *       OUT     6
                   *
                   *-----------------------------------------------------------------
                   *
                   * FIRST WAIT AFTER SENDING EOF, LAST DATA CHAR
                   *
                   *       LOC     TDAT12
           050D    TDAT12 EQU     X'050D'
                   *       WAIT
     135A  3601
                   *
                   *       IN      5
                   *       ST      X'2E'
     135B  2551
                   *       AND     =X'1'              STRIP TO UNDERRUN BIT
     135C  2E93
                   *       BZF     XUR                BRANCH IF UNDERRUN
     135D  01F2
                   *       LOC     TDAT6
           0515    TDAT6  EQU     X'0515'
                   *       GNB
     135E  4602
                   *       LD      =0
     135F  9000
                   *       ST      X'38'              CLEAR CHAR COUNT
     1360  5138
                   *-------------
                   *
                   * TEST IF CONTROL FLAG IS SET TO TURN ON RCV AFTER
                   * A SPECIFIED XMIT FRAME.
                   *
                   *       LD      X'3E'              GET FLAG
     1361  503E
                   *       AND     =4                 STRIP TO DELAY TURN ON OF RCV
     1362  9304
                   *       BZT     TDAT11
     1363  E212
                   *
                   * TURN ON RCV
                   *
                   *       LD      X'19'              ACTION FRAME
     1364  5019
                   *       C       X'3B'              COMPARE WITH ACTUAL
     1365  523B
                   *       BEF     TDAT11             B = NOT THIS FRAME
     1366  F10C
                   *       LD      =20                DELAY TO ALLOW LAST CHAR'S
     1367  9014
                   *       BS      CCPDLY             TO GET TO RCVR
     1368  F03B
                   *       LD      X'14'              LR2 CONTROL
     1369  5014
                   *       OR      =2                 PUT IN RCV BIT
     136A  9402
                   *       ST      X'14'
     136B  5114
                   *       OUT     2
                   *-------------
                   *
                   * TEST FOR LAST FRAME XMITTED.
                   *
                   *       LOC     TDAT11
           0531    TDAT11 EQU     X'0531'
                   *       BLBF    TDAT7
     136C  32F4
                   *-----------------------------------------------------------------
                   *
                   * END OF TRANSMISSION - SHUT OFF XMIT
                   *
                   *       LOC     TEND
           0533    TEND   EQU     X'0533'
                   *       LD      X'14'              GET LR2 CONTROL
     136D  1050
                   *       AND     =X'FE'             STRIP OFF XMIT ON
     136E  1493
                   *       ST      X'14'
     136F  FE51
                   *       OUT     2
     1370  1432
                   *       JUMP    LAST1
     1371  E6FE
                   *-------------
                   *       LOC     TONER
           053D    TONER  EQU     X'053D'
                   *       LD      =14                ERROR CODE
     1372  C690
                   *       JUMP    EREND              ERROR, DELAYED XMIT DIDN'T FUNCTION
     1373  0EE6
     1374  FE5B           DC      EREND-X'0541'
                   *-----------------------------------------------------------------
                   *
                   * INTERFRAME STATE
                   *
                   *       LOC     TDAT7
           0542    TDAT7  EQU     X'0542'
                   *       LD      X'3E'              GET FLAG
     1375  503E
                   *       AND     =X'20'             CHECK FOR INTERFRAME DELAY
     1376  9320
                   *       BZT     TDAT8              B = NO DELAY
     1377  E210
                   * INTERFRAME DELAY ROUTINE. NOTE THAT 'PAUSE FEATURE' OF MLCP
                   * PREVENTS THIS DELAY FROM "HOGGING" OUT RCV.
                   *       LD      =20
     1378  9014
                   *       BS      CCPDLY             DELAY 100 MS
     1379  F019
                   *
                   * ROUTINE TO ISSUE RCV RE-SYNC IF SPECIFIED.
                   * THIS OCCURS AFTER DELAY BETWEEN FRAMES
                   *
                   *       LD      X'3E'              GET FLAG
     137A  503E
```

```
                          *        AND      =X'40'                RCV RE-SYNC
 137B   9340
                          *        BZT      TDAT8                 B = NO RE-SYNC
 137C   E206
                          *        OUT      3                     RE-SYNC
                          *
                          *        XOR      X'3E'                 RESET BIT
 137D   3355
                          *        ST       X'3E'                 STORE BACK
 137E   3E51
                          *        LOC      TDAT8
        0557              TDAT8    EQU      X'0557'
                          *        JUMP     SFRM
 137F   3EE6
 1380   FEF5                      DC       SFRM-X'0559'
                          *-------------
                          *
                          * UNDERUN OCCURED
                          *
                          *        LOC      XUR
        055A              XUR      EQU      X'055A'
                          *        LD       X'30'
 1381   5030
                          *        OR       =X'20'
 1382   9420
                          *        ST       X'30'                 STORE UNDERUN BIT IN CP STATUS
 1383   5130
                          *        ST       X'20'                 SET BIT IN STATUS
 1384   5120
                          *        B        TDAT6
 1385   E0B2
                          *-------------
                          *
                          * DELAY ROUTINE
                          *
                          * CCPDLY
                          *        LOC      CCPDLY
        0564              CCPDLY EQU        X'0564'
                          *        ST       X'39'                 STORE COUNT
 1386   5139
                          * CCPD1
                          *        LOC      CCPD1
        0566              CCPD1    EQU      X'0566'
                          *        LD       =0
 1387   9000
                          *CCPD2
                          *        LOC      CCPD2
        0568              CCPD2    EQU      X'0568'
                          *        DEC
                          *        AND      =X'FF'
 1388   0593
                          *        AND      =X'FF'
 1389   FF93
                          *        AND      =X'FF'
 138A   FF93
                          *        BZF      CCPD2
 138B   FFF2
                          *        LD       X'39'
 138C   F850
                          *        DEC
 138D   3905
                          *        ST       X'39'
 138E   5139
                          *        BZF      CCPD1
 138F   F2EF
                          *        RET      END                   OF ROUTINE
                          *        NOP
 1390   0600
                          *        NOP
                          *
                          *        NOP
 1391   0000
                          *        NOP
                          *-------------
                          *
                          *        LOC      XDLAY
        057D              XDLAY    EQU      X'057D'
                          *        BS       CCPDLY
 1392   00F0
                          *        JUMP     TDAT3
 1393   E6E6
 1394   FF27                       DC       TDAT3-X'0581'
                          *-------------
                          *        LOC      XDLAY1
        0582              XDLAY1 EQU        X'0582'
                          *        BS       CCPDLY                DELAY
 1395   F0E1
                          *        JUMP     TDAT13
 1396   E6FF
                          *        NOP
 1397   7300
                          *        NOP
                          *        NOP
 1398   0000
                          *        NOP
        1399              CCP3     EQU      $
                          *        NOP
 1399   0000
                          *-----------------------------------------------------------------------
                          *-----------------------------------------------------------------------
                          *-----------------------------------------------------------------------
                          *
                          *
                          *
                          *
                          * DATA SET STATUS CHANNEL PROGRAM
                          *
                          *        ORG      X'200'
                          * DS0
                          *        LOC      DS0
        0200              DS0      EQU      X'0200'
        139A              CCP4     EQU      $
                          *        NOP
```

```
                         *          NOP
       139A  0000
                         *          NOP
                         *          NOP
       139B  0000
                         *          NOP
                         *          NOP
       139C  0000
                         *          LD       =0
       139D  9000
                         *          BS       GSTAT                OUTPUT CONTROL AND INPUT STATUS
       139E  F032
                         *          C        =0
       139F  9200
                         *          BET      DS1
       13A0  E103
                         *          BS       BADS                 BAD STATUS
       13A1  F025
                         *
                         * DS1
                         *          LOC      DS1
             0210        DS1     EQU      X'0210'
                         *          WAIT
                         *          LD       =X'80'
       13A2  0190
                         *          BS       GSTAT                OUTPUT CONTROL AND INPUT STATUS
       13A3  80F0
                         *          C        X'3E'                10 FOR BHCLA1, 90 FOR BHCLA2
       13A4  2752
                         *          BET      DS2
       13A5  3EE1
                         *          BS       BADS                 BAD STATUS
       13A6  03F0
                         *
                         * DS2
                         *          LOC      DS2
             021B        DS2     EQU      X'021B'
                         *          WAIT
       13A7  1A01
                         *          LD       =X'40'
       13A8  9040
                         *          BS       GSTAT                OUTPUT CONTROL AND INPUT STATUS
       13A9  F01C
                         *          C        X'3F'                E0 FOR BHCLA1, 60 FOR BHCLA2
       13AA  523F
                         *          BET      DS3
       13AB  E103
                         *          BS       BADS                 BAD STATUS
       13AC  F00F
                         *
                         * DS3
                         *          LOC      DS3
             0226        DS3     EQU      X'0226'
                         *          WAIT
                         *          LD       =X'3C'
       13AD  0190
                         *          BS       GSTAT                OUTPUT CONTROL AND INPUT STATUS
       13AE  3CF0
                         *          C        =X'0'
       13AF  1192
                         *          BET      DS4
       13B0  00E1
                         *          BS       BADS                 BAD STATUS
       13B1  03F0
                         *
                         * DS4
                         *          LOC      DS4
             0231        DS4     EQU      X'0231'
                         *          WAIT
       13B2  0401
                         *          B        DS0
       13B3  E0CD
                         *
                         * ROUTINE FOR BAD STATUS
                         *
                         * BADS
                         *          LOC      BADS
             0234        BADS    EQU      X'0234'
                         *          LD       X'10'
       13B4  5010
                         *          OR       =X'4'
       13B5  9404
                         *          ST       X'10'
       13B6  5110
                         *          RET
                         *
                         * ROUTINE TO OUTPUT CONTROL AND INPUT STATUS
                         *
                         * GSTAT
                         *          LOC      GSTAT
             023B        GSTAT   EQU      X'023B'
                         *          OUT      2                    OUTPUT CONTROL
       13B7  0632
                         *          NOP
                         *          LD       =0
       13B8  0090
                         *          LOC      STDLY
             023F        STDLY   EQU      X'023F'
                         *          DEC
       13B9  0005
                         *          BZF      STDLY                DELAY
       13BA  F2FE
                         *
                         *          NOP
                         *          NOP
       13BB  0000
                         *          NOP
                         *          NOP
       13BC  0000
                         *          IN       5                    INPUT STATUS
                         *          ST       X'2E'
       13BD  2551
                         *          AND      =X'F7'               STRIP OFF ADAPTER READY
```

```
006788   13BE   2E93
006791                                   *        RET     RETURN
006792   13BF   F706
006793                                   *        NOP
006794                                   *        NOP
006795   13C0   0000
006796          13C1                     CCP5     EQU     $
006797                                   *
006798                                   *
006799   1800                                     ORG     ZERO+X'1800'
006800   1800                            SDB      RESV    X'400'                 SEND BLOCK BUFFER
006801   1C00                            RTB      RESV    X'400'                 RETURN BLOCK AREA
006802   2000                                     ORG     ZERO+X'2000'
006803                                   *
006804                                   ** ROUTINE TO PRINT TEST LABLE
006805                                   *
006806                                   PLB      CALL    ZV$BRK
         2000   FBF0 0001
         2002   D380 0000        X
006807   2004   8980 0000        X                CMZ     <ZV$BKF
006808   2006   0980 0105                         BNE     <RESTRT                IF BREAK, RESTART
006809                                   *
006810   2008   89C0 F176                         CMZ     PFLAG
006811   200A   0981 0002                         BNE     PLB1
006812   200C   8384                              JMP     $B4                    EXIT
006813   200D   9800 11EA                PLB1     LDR     $R1,<ERMG+1            GET TEST LABLE
006814   200F   9F00 201B                         STR     $R1,<PLB3
006815                                            CALL    ZV$TC,PLB2             PRINT LABLE
         2011   FBC0 0003
         2013   D380 0000        X
         2015   0F80
         2016   2018
006816   2017   8384                              JMP     $B4
006817                                   *
006818   2018   5445 5354 2020           PLB2     TEXT    'TEST  '
006819   201B   2020 2424                PLB3     TEXT    '  $$'
006820                                   *
006821                                   *
006822                                   * SUBROUTINE SAVE AREA'S
006823                                   *
006824   201D   0000                     SAV1     RESV    9+7*$AF,0              SDATA,RDATA
006825   202D   0000                     SAV2     RESV    9+7*$AF,0              GENITZ,CHCT,TEST,MCCB
006826   203D   0000                     SAV3     RESV    9+7*$AF,0              CGSCH
006827   204D   0000                     SAV4     RESV    3+3*$AF,0              RPVLU
006828   2053   0000                     SAV5     RESV    9+7*$AF,0              ERROR
006829   2063   0000                     SAVMAJ   RESV    9+7*$AF,0
006830   2073   0000                     SAV      RESV    9+7*$AF,0
006831   2083   0100                              END     DCMS3,<STRT
0000  ERR COUNT
  TITLE DCMS3, 'REV A'
          $AF          750B    1186B    2611C    2898C    2911C    3795     3982B    4012C    4058     4139C
                      4201B    4204B    4245B    4292B    4297B    4475     4477     6824     6825     6826
                      6827     6828     6829     6830
          $B1         1128B    1165     1169     1173     1251     1253C    1257C    1325     1326     1337
                      1339     1342     1570     1571     1582     1584     1587     1592     1595     1913
                      1915     1918     1920     1922     1924     1926     2442     2445     2447     2449
                      2451     2453     2456     2610     2611C    3062     3084     3124     3125     3221B
                      3229B    3395B    3468     3470C    3734     3735C    3736     3760     3761C    3762
                      3773     3774C    3775     3785     3786C    3787     4036     4043     4045     4058
                      4061     4179     4180     4183     4185     4207B    4221     4223     4225     4227
                      4258     4261
          $B2          717      720      947B     969      970C     973B     994      995C     999B    1020
                      1021C    1025B    1046     1047C    1051B    1072     1073C    1077B    1260     1261C
                      1266B    1294     1295C    1300B    1332     1333     1355     1356C    1359B    1449
                      1450C    1454B    1491     1492C    1496B    1531     1532C    1536B    1612     1613C
                      1616B    1723     1724C    1728B    1759     1760C    1764B    1796     1797C    1801B
                      1823     1824C    1828B    1850     1851C    1855B    1877     1878C    1882B    1952
                      1953C    1957B    2248     2249C    2253B    2283     2284C    2288B    2318     2319C
                      2323B    2365     2366C    2367B    2409     2410C    2415B    2477     2478C    2488B
                      2581B    2701B    2788B    2838B    2964B    3545     3556     3562C    3566C    3569
                      3571C    3576C    3586C    3602C    3611     3615C    3620C    3639     3641C    3693C
                      3696     3700     3706     3712     3717     3722     3724     3728     3730     3732
                      3734     3738     3752     3760     3764     3767     3773     3777     3780     3785
                      3789     3792     3809     4285     4291
          $B3          992B    1044B    1070B    1145B    1148B    1154B    1296B    1354B    1447B    1489B
                      1529B    1610B    1721B    1757B    1794B    1821B    1848B    1875B    1950B    2246B
                      2281B    2316B    2574B    2585B    2694B    2703B    2780B    2791B    2832B    2843B
                      2897     2898C    2907     2909     2910     2911C    2956B    2982B    3005B    3070B
                      3164B    3358B    3404     3411     3419     3424C    3475B    3793B    3942B    3944B
                      3957B    3959B    3963B    3966B    4025B    4028B    4165B    4258     4265B
          $B4          718      719C     840B     971B     997B    1023B    1049B    1075B    1118B    1120B
                      1122B    1158B    1161B    1171B    1184B    1263B    1298B    1357B    1451B    1493B
                      1533B    1614B    1725B    1761B    1798B    1825B    1852B    1879B    1954B    2250B
                      2285B    2320B    2412B    2479B    2483B    2487B    2577B    2597B    2601B    2616B
                      2621B    2626B    2634B    2643B    2697B    2727B    2728B    2740     2741     2783B
                      2834B    2896     2897     2899     2909     2910     2925B    2926B    2933B    2959B
                      2986B    2995B    3011B    3020B    3029B    3074B    3083B    3159B    3238B    3251B
                      3263B    3274B    3286B    3293B    3300B    3307B    3314B    3330B    3350B    3354B
                      3384B    3393B    3524B    3645B    3677B    3694B    3736     3737C    3762     3763C
                      3775     3776C    3787     3788C    3819     3822B    3835B    3840B    3843B    3847B
                      3850B    3855B    3864     3865     3867B    3873B    3887     3888     3892B    3899B
                      3912B    3919B    3922B    3933B    3973B    3980B    3985B    4141B    4147B    4157B
                      4162B    4187B    4191B    4197B    4229B    4233B    4243B    4250B    4268B    4285
                      4286     4287     4288C    4290B    4295B    4299C    4301B    6812B    6816B
          $B5          788      791      793C    2801B    2802B    3546     3550     3582     3590     3605
                      3626     3882     4138     4139C    4179     4182C    4221     4222C    4288C    4299C
                      4319B    4352B
          $B6          751B     842B     857B     859B     861B     863B     873B     875B     877B     879B
                       881B     883B     885B     889B     891B     893B     895B     898B     900B     902B
                       904B     906B     908B     910B     913B     915B     917B     919B     924B     959B
                       985B    1011B    1037B    1063B    1089B    1116B    1176B    1187B    1192B    1279B
                      1312B    1344B    1471B    1514B    1554B    1631B    1747B    1776B    1814B    1841B
                      1868B    1895B    1977B    2274B    2309B    2344B    2355B    2359B    2381B    2428B
                      2506B    2596B    2608B    2624B    2630B    2638B    2654B    2670B    2684B    2722B
                      2739B    2763B    2769B    2887B    2892B    2920B    2990B    2999B    3014B    3024B
                      3033B    3048B    3078B    3087B    3456B    3846B    3870B    3902B    3915B    3950B
                      3983B    4010     4011     4012C    4124B    4202B    4205B    4246B    4293B    4298B
          $R1          673      674C     681      682C     683      684      701      702C     703C     704C
                       706      707C     714      715B     716C     720      721B     722      724      726
                       727      730      734      735C     748      749      758      759      760      763C
                       764      765C     767      768C     787      790      792      793C     804      805
                       806      812      814      820      824      828      836      837C     850      851C
                       865      866      868      869C     932      933C     943      944C     965      966C
```

```
                     990       991C     1017      1018C     1042      1043C     1068      1069C     1109      1110C
                    1113      1114B     1139      1140C     1142      1157C     1173      1194C     1195      1245
                    1246C     1252      1253C     1254      1257C     1258      1290      1291C     1292      1293C
                    1326      1328C     1329      1330C     1331      1333      1334C     1335      1336C     1337
                    1338C     1339      1340      1341C     1342      1343B     1346      1347C     1440      1441C
                    1442      1443C     1481      1482C     1483      1484C     1521      1522C     1523      1524C
                    1571      1572C     1573      1574C     1575      1577      1578C     1580      1581C     1582
                    1583C     1584      1585      1586C     1587      1588B     1589      1590C     1592      1593
                    1594C     1595      1597C     1598      1599C     1713      1714C     1715      1716C     1754
                    1755C     1789      1790C     1791      1792C     1816      1817C     1818      1819C     1843
                    1844C     1845      1846C     1870      1871C     1872      1873C     1912C     1915      1918
                    1920      1922      1924      1926      1928C     1935      1936C     1947      1948C     1974
                    2241      2242C     2243      2244C     2276      2277C     2278      2279C     2311      2312C
                    2313      2314C     2350      2351C     2353      2354B     2386      2387C     2391      2392
                    2395      2396C     2397      2398C     2399      2404C     2438      2439C     2440C     2445
                    2447      2449      2451      2453      2456      2463      2466C     2476C     2501      2504
                    2505B     2589      2591      2603      2604      2618      2619B     2639      2641      2657
                    2658      2665      2667      2677C     2679      2680      2681C     2709      2710      2716
                    2717      2741      2742      2743C     2744      2745C     2751      2753      2754C     2764
                    2765C     2771      2792      2793      2794      2795B     2796C     2797C     2810      2811C
                    2815      2817C     2818C     2847      2848C     2902      2906C     2907      2922      2923
                    2971      2973      2978      2979C     2980      2981C     3059      3060B     3064      3066B
                    3068C     3093      3094C     3121      3122B     3123      3125      3132      3134B     3137
                    3138      3139C     3143      3144B     3145      3146B     3147      3148      3149      3150C
                    3152      3153B     3154      3155      3156C     3248      3249      3319      3323      3324C
                    3339      3340C     3341C     3343      3344      3387      3388      3406      3407      3412
                    3413C     3416      3417      3418C     3420C     3422      3423      3424C     3425      3426
                    3427C     3428B     3432      3433      3434C     3441      3442      3444      3445      3446C
                    3450      3451      3452C     3467C     3470C     3472      3480      3484C     3492      3493
                    3494      3495      3515      3517C     3521C     3549      3552      3553      3556      3557
                    3562C     3566C     3569      3571C     3576C     3577      3579C     3581C     3586C     3587C
                    3602C     3611      3615C     3620C     3636      3668      3671      3680C     3682C     3684
                    3686      3696      3697      3698      3699C     3700      3701C     3703      3704      3705C
                    3706      3707      3708C     3709      3710      3711C     3712      3713      3714      3715
                    3716C     3717      3718      3719      3720C     3722      3723C     3724      3725      3726
                    3727C     3728      3729C     3730      3731C     3732      3733C     3738      3739C     3743
                    3744C     3752      3753C     3754C     3755      3764      3765C     3766C     3777      3778C
                    3779C     3780      3781C     3782C     3783      3789      3790C     3791C     3792      3812
                    3813      3815      3819      3883C     3884C     3885      3886C     3887      3888      3889C
                    3890C     3909      3910C     3916      3918      3926      3927C     3928      3929      3940
                    3941C     3952      3955      3956C     3961C     3962C     3990      3991C     3992      3993
                    3994C     3998      3999C     4000      4001C     4009      4021C     4022      4030C     4031C
                    4037      4038C     4039      4052C     4053      4129      4130      4136      4137      4140C
                    4142C     4143C     4144      4145C     4185      4186      4225      4226C     4227      4228
                    4259C     4261      4262C     4307C     4310      4313      4314      4316      6813      6814C
        $R2          713C      720       785       791       793C      794B     1166      1167      1255      1259B
                    1323C     1326      1337      1339      1342      1348C     1371      1568C     1571      1582
                    1584      1587      1592      1595      1596C     1597C     1600C     1601C     1602      1606C
                    1628      1915      1916B     1917C     1918      1919C     1920      1921C     1922      1923C
                    1924      1925C     1926      1927C     1939      1940      1943      1944C     1945      1946C
                    2393      2394      2395      2400      2401B     2402      2443      2457C     2659      2660
                    2661      2662      2663      2678      2685B     2849C     2856C     2860C     2866C     2896
                    2902      2908C     2972C     2975      2976C     3044C     3045      3046C     3061C     3064
                    3084      3120C     3125      3127      3128      3129C     3133C     3135C     3136C     3320
                    3321      3405      3409      3436B     3469C     3470C     3471C     3484C     3495      3496B
                    3497      3503B     3556      3560      3561      3562C     3565      3566C     3569      3570
                    3571C     3574      3575      3576C     3585      3586C     3593      3594      3595      3599C
                    3601      3602C     3609      3612      3616C     3626      3627      3628      3629      3633
                    3634      3636      3639      3641C     3642      3663      3672C     3673      3674      3701C
                    3702      3740      3741      3751C     3767      3768C     3769C     3771      4014      4015B
                    4022      4023B     4024C     4050      4051C     4056      4057C     4183      4184C     4261
                    4263B     4269      4287      4296
        $R3          839C     1119C     1126B     1127C     1134C     1138      1139      1141      1181C     1182B
                    1183C     1189      1197      1322C     2576C     2583C     2615C     2687      2688      2696C
                    2726C     2772      2773      2782C     2790C     2804      2805C     2816      2817C     2820
                    2821      2833C     2840C     2935      2936      2958C     2969C     2970      2971      2977
                    2994C     3010C     3019C     3028C     3044C     3049      3082C     3091      3092      3158
                    3220C     3280C     3329C     3334C     3381B     3382C     3390C     3402C     3403      3424C
                    3440      3482C     3485      3486C     3490      3500      3516      3517C     3518      3519C
                    3520      3521C     3522      3523C     3808      3810      3812      3817C     3821      3832
                    3833      3834      3953B     3954C     3992      4008C     4329      4330      4335      4341
                    4347
        $R4          699       700C      789       791       792      1180      1185      1333      1334C     1577
                    1578C     2445      2446C     2447      2448C     2449      2450C     2451      2452C     2453
                    2454C     2455B     2456      2457C     2661      2662      2663      2664C     2798      2799
                    2800C     2865C     2867      2869      2873      2876      2879      2913C     3383C     3390C
                    3391      3394C     3552      3554      3555C     3557      3607      3608      3612      3624
                    3625B     3629      3630      3631C     3638C     3639      3640      3641C     3669      3670
                    3680C     3682C     3685      3686      3770C     3833      3842      3844      3866      3868
                    3898      3900      3911      3913      3979      3981      4010      4011      4043      4044
                    4046      4048C     4061      4146      4151      4161      4163      4180      4181C     4223
                    4224C     4242      4244      4267      4269      4289      4291      4294      4296      4307C
                    4308      4309C     4310      4311      4312      4315      4316      4317C     4339      4346
        $R5          727       728       740C      749       752       753       754C      756       765C      776
                    1172      1174      1185      1191B     2594      2606      2622      2628      2636      2651
                    2667      2668      2680      2682      2720      2737      2747      2748      2760      2766C
                    2767      2771      2860C     2861      2885      2890      2918      2988      2997      3012
                    3022      3031      3076      3085      3485      3505      3508      3611      3614      3615C
                    3619      3620C     3634      3635C     3673      3679      3864      3865      3868      3893
                    3900      3945C     3948      3960C     3967      3968      3969C     3971      3981      4029
                    4030C     4045      4046      4059      4062B     4151      4153C     4163      4200      4203
                    4244      4286      4291      4334C     4335      4336C     4340C     4341      4342      4343
                    4349
        $R6          688C      692C      698C     1173      1174      1190C     2590C     2593      2594      2602
                    2606      2620      2622      2627      2628      2635      2636      2652      2665      2666
                    2668      2679      2682      2707C     2715      2719      2720      2734      2736      2737
                    2761      2762      2767      2852C     2854      2857C     2862      2864C     2865C     2871C
                    2875      2878      2881      2884      2885      2889      2890      2899      2900      2901C
                    2903      2904      2919      2927C     2929      2987      2988      2996      2997      3009
                    3012      3021      3022      3030      3031      3075      3076      3084      3085      3505
                    3506      3510B     3947C     3948      4328C
        $R7          690       691       693       694C      695       696       697C      778      2653      2708
                    2712      2717      2746      2750      2751      2851      2853      2928      2931      3481
                    3506      3509      3511      3514C     3516      3664      3665      3668      4331      4332
                    4333      4334C     4337      4338      4339      4340C     4344      4345      4346      4347
                    4348      4349      4350C
2694    ABUND       1616B     1764B
2736    ABUND1      2733B
2697    ABUND2      2774B
2703    ABUND3      2699B
2720    ABUND4      2718B
2737    ABUND5      2735B
2700    ABUND6      2698B
```

```
2716   ABUND7    2714B
2713   ABUND8    2711B
2751   ABUND9    2749B
3191   ACTFLG    2672      2713      2732      3723C     3740
3556   ADPR      3563B
4378   ADSTR
 789   ALL        794B
4376   ALLONE
4466   ATLT       702C      703C      704C      765C     1139      2971      3809      3812
 738   BADINP     736
6736   BADS      6642      6643      6644      6668      6669      6675      6693      6694      6695      6719
                 6720      6726
4403   BASE      2396C     2406
3503   BITLUP    3510B     3512B
4327   BPS       2802B
 925   BYPSS      922B
3667   BYTTZ     3688B
4408   C0         705      2362
4377   C16
4416   C4        2841
4410   C70       1019
4409   C8         705       967
4399   CBLOOP     801C      809C     1113      3059      3143
1118   CBLP1     1114B
1120   CBLP2     1198B
1194   CBLP4     1191B
1158   CBLP5     1196B
1184   CBLP6     1182B
1128   CBLP7     1126B
1115   CBLP8     1121B
1157   CBLP9
1180   CBLP-3    1175B
1109   CBLP-E     863B
3760   CCBA      3756B
4504   CCP1      3222      3223
5592   CCP2      3223      3230      3231
6599   CCP3      3231
6615   CCP4      1129      1130
6796   CCP5      1130
6522   CCPD1     6558      6559      6560
6529   CCPD2     6545      6546      6548
6515   CCPDLY    6354      6355      6356      6445      6446      6447      6575      6576      6578      6586
                 6587      6588
2915   CDATA     2898C     2911C
3569   CFP       3551B
3581   CFP1      3573B
3832   CGSCH     1184B     3843B     3867B     3899B     3912B     3980B     4147B     4162B     4243B     4268B
                 4290B     4295B
4367   CHAN      2583C     2687      2772      2790C     2820      2840C     2935      2969C     2977      3049
                 3091      3158      4008C
3863   CHCT      1158B     3330B     3350B     3384B     4191B     4233B
1154   CHLPX     1147B
1148   CHLPY     1143B
3484   CHLUP     3498      3504B
4369   CHRCNT    2797C     4329
4371   CHSZ       946C     2364C     3123      3548C     3579C     3597      3599C     3624      3631C     3633
3872   CHZ       3869B
4417   CM1       2841
2648   COMP      2611C     2637B     2642B
2757   COMP1
 726   CON1
 748   CON2       729B
 785   CON3       721B
 758   CON4       779B
4445   CONT1      788      4289
4454   CONT10    1180      4161
4455   CONT11
4456   CONT12    3979
4446   CONT2     3911
4447   CONT3     4294
4448   CONT4     4242
4449   CONT5     4267
4450   CONT6     3898      4146
4451   CONT7     3866
4452   CONT8
4453   CONT9     3842
4370   COUNT     2439C     2504
2442   CRC-LP    2502B
2522   CRCTST    2442
4394   CRLF      4037      4056
1171   CUR       1168B
4379   DADD       706       716C
3460   DASH      3430
4479   DATE      4477
3624   DCHSZ     3606B
       DCMS3        1      6831
4375   DEVID      754C      772      1140C     1166
2233   DEX       1348C     1371      1596C     1628      1928C     1974      2476C     2501
4380   DFLT      3854
4413   DIV1       694C     4308
4414   DIV2       697C     4311
4415   DIV3       698C     4313
4128   DLAY      2616B
4150   DLAY1     4154B
4138   DLAY2     4131B
4135   DLAYLG    1161B     2597B     2727B     2925B     2926B     3847B
4790   DLOOP
3649   DMASK     2663      3634
4120   DMPLOC    3941C     3943      3956C     3958      3962C     3964      4024C     4026
4766   DOCFG     4750      4751      4753
4381   DRNG      3854
6614   DS0       6728      6729      6730
6648   DS1       6638      6639      6640
6673   DS2       6664      6665      6667
6699   DS3       6689      6690      6691
6724   DS4       6715      6716      6718
1201   DS-SB     1165
1208   DS-SB1    1169
3101   DSC       3064
3103   DSC1      2976C
3978   DSSTA     1171B
4361   DUMMY     3239      3252      3264      3275      4188      4230      4361
3565   EADF      3558B
```

```
3619  ELCF     3613B
4368  ELPS     2800C   3890C   3916
1631  END-J    1588B
2504  ENDCRC   2455B
6113  EOF      6067    6068    6069    6092    6093    6097
6166  EOF1     6140    6141    6147
6188  EOF2     6162    6163    6168
5154  EOFP     5546    5547    5548
5194  EOFR     5189    5190    5191
4487  EQ        771
4913  ER-5     4958    4959    4961
4468  ERAR     2648    2649    2651    2652    2653    2757    2758    2760    2761    2762
               2915    2916    2918    2919
5555  EREND    4519    4520    4521    4610    4611    4612    4639    4640    4641    4666
               4667    4668    4681    4682    4683    4690    4700    4701    4702    4709
               4728    4729    4730    4737    4919    4920    4921    4925    5007    5008
               5009    5020    5069    5070    5071    5222    5223    5224    5361    5362
               5363    5370    5389    5390    5391    5607    5608    5609    6417    6418
               6419
4390  ERF
4483  ERMG      735C    837C    944C    966C    991C   1018C   1043C   1069C   1110C   1246C
               1291C   1347C   1441C   1482C   1522C   1590C   1714C   1755C   1790C   1817C
               1844C   1871C   1917C   2242C   2277C   2312C   2351C   2387C   2454C   2457C
               2603    2709    2747    3387    3406    3441    3991C   3999C   4013    6813
4013  ERR1     4012C
4122  ERR5     4063B
4021  ERR6     4015B
4363  ERRCD    3945C
3989  ERRDB    1176B   1192B   2596B   2608B   2624B   2630B   2638B   2654B   2670B   2684B
               2722B   2739B   2763B   2769B   2887B   2892B   2920B   2990B   2999B   3014B
               3024B   3033B   3078B   3087B   3950B
3997  ERRMB     751B   1187B   3846B   3870B   3902B   3915B   3983B   4202B   4205B   4246B
               4293B   4298B
4008  ERROR    3995B   4002B
4022  ERRP1    4032B
4163  ERRP2    4164B
1977  ETFR     1916B
4389  EXST     1443C   1484C   1524C   1716C   1792C   1819C   1846C   1873C   1919C   2244C
               2279C   2314C   2851
3470  FACDT1   3473B
3466  FACDTA    992B   1044B   1070B   1296B   1354B   1447B   1489B   1529B   1610B   1721B
               1757B   1794B   1821B   1848B   1875B   1950B   2246B   2281B   2316B
3854  FDFLT    3314B
3807  FLN       840B   1120B   2577B   2697B   2783B   2834B   2959B   3840B
3127  FMASK    3122B
2852  FNULL    2858B
 686  FREQ
3458  FREV     3401
3193  FRMCNT
4365  FRST      665     678     680C
3480  GCRC     2483B
3644  GDAT7
3639  GDAT8    3643B
3839  GENITZ   1122B   3159B   3393B
3544  GHEAD     971B    997B   1023B   1049B   1075B   1263B   1298B   1357B   1451B   1493B
               1533B   1614B   1725B   1761B   1798B   1825B   1852B   1879B   1954B   2250B
               2285B   2320B   2412B   2479B
1360  GLR6     1338C   1355
1361  GLR7     1328C
1362  GLR8     1330C
3500  GRC1     3491B   3496B
3505  GRC2     3503B
4061  GREG     4055B
6755  GSTAT    6630    6631    6632    6656    6657    6659    6681    6682    6683    6707
               6708    6710
4433  HCLS-1   2817C
4434  HCLS-2
 857  HDTSA     858     860     862     864     874     878     880     882     884     887
                890
4382  HEAD      968C    996C   1022C   1048C   1074C   1262C   1293C   1336C   1340    1341C
               1448C   1490C   1530C   1581C   1585    1586C   1722C   1758C   1795C   1822C
               1849C   1876C   1951C   2247C   2282C   2317C   2411C   2466C   3121    3552
               3572    3593    3607
4411  HRTZ      682C    687     691    3929    4129    4136
 919  HSPD      870B
4486  IDMSG     769
4383  IFLG      670C   2458C   2467C   2471C   2485    2639    2640C
3194  IFM      2979C   3003C   3716C   3726    3727C
4373  IMASK
4388  INBUF
4161  INBYTE   3944B   3959B   3966B   4028B
3908  INRNG
4241  INXT     2621B   2626B   2728B   2933B   3011B   3892B   4248
3847  ITZ      3845B
3662  IVRT     2487B   2643B
5227  JMP2     5025    5026    5030
 730  JP1       723B    725B
 734  JP2       715B    731B    761B    777B
 776  JP3       757B
4478  LAF1     4475
4510  LAST     5298    5299    5300    5309
5599  LAST1    6405    6406    6407    6412
3605  LCFP     3591B
3611  LCFP1    3617B
3170  LCT1     3165
1222  LCT10    1149
2577  LCT2     2689B
4269  LCT3     4270B
4261  LCT4     4271B
4267  LCT5     4263B
1231  LCT6     1155
3151  LCT7     3144B   3146B
2583  LCT8     2579B
1214  LCT9     1146
5174  LFRM     5539    5540    5541    5545
2824  LINE     2806
4126  LINEP    4016
3810  LN2      3818B
3820  LN3      3811B
3819  LN4      3814B   3816B
4366  LNBM     3994C   4001C   4014    4017
3116  LOAD     2585B   2703B   2791B   2843B   2982B   3005B   3070B
4066  LOCTAB   4022
```

```
4387   LOOP      852C    926C      2357      2393      3038
 872   LOSPD     867B
4812   LP1
3158   LPT3     3153B
3136   LPT6     3130     3134B
1017   LPT-BA    875B
2574   LPTS      973B    999B      1025B     1051B     1077B     1266B     1300B     1359B     2415B     2488B
2594   LPTS1    2592B
2580   LPTS2    2578B
2609   LPTS3    2605B    2607B
3952   LPTS4    3949B
 965   LPTS-A    859B
 990   LPTS-B    873B
1042   LPTS-C    877B
1068   LPTS-D    879B
3185   LR2CFG   3068C    3094C     3147      3150C     3154      3156C     3720C
3186   LR6RCV   3699C
3187   LR7XMT   3705C
5574   LUP      5578     5579      5580
4833   LUP2     4822     4823      4824      4926      4927      4929      4931      4932      4933
4384   MASK      803     2609C     2648      2664C     2666      2756C     2757      2912C     2915      3635C
                3640
4296   MCB2
4284   MCCB     3238B    3251B     3263B     3274B     3286B     3293B     3300B     3307B     4187B     4229B
4480   MESG1     711
4481   MESG2     802
 844   MESG3     838
3096   MESG4    3043
4482   MESG5     686
4420   MLC-FR   3452C
4404   MODE     2406
4401   MSBS     2809     2815      4336C     4343      4350C
3362   MSKTBL   3124
1950   MTS1     1938B    1941B
1521   MUL-IA    893B    1552
1440   MULT-H    889B
1481   MULT-I    891B    1512
1713   MULT-L    898B
4923   NEXT     4909     4910      4911
 740   NODEVF    718
 834   NOGO      831
 856   NOP       673      850
5028   NOTESM   4999     5000      5001
5012   NOTEST   4987     4988      4989
4421   ODDFLG   2655     3136C
4773   OUTIT    4762     4763      4768
4094   OUTTAB   4030C    4036
1981   PAR      1913
 974   PAR1      969
1729   PAR10    1723
1765   PAR11    1759
1802   PAR12    1796
1829   PAR13    1823
1856   PAR14    1850
1883   PAR15    1877
1958   PAR16    1952
2254   PAR17    2248
2289   PAR18    2283
2324   PAR19    2318
1000   PAR2      994
2416   PAR20    2409
2368   PAR21    2365
1959   PAR22    1936C    1948C
1026   PAR3     1020
1052   PAR4     1046
1078   PAR5     1072
1267   PAR6     1260
1301   PAR7     1294
1455   PAR8     1449
1497   PAR9     1491
1537   PAR91A   1531
3183   PARBYT   2659     3711C
3795   PARPTR    970C    995C      1021C     1047C     1073C     1261C     1295C     1356C     1450C     1492C
                1532C    1613C     1724C     1760C     1797C     1824C     1851C     1878C     1953C     2249C
                2284C    2319C     2366C     2410C     2478C     2740      3546      3693C
3793   PASEND   3757B    3772B     3784B
 938   PASMSG    935
3693   PASS     2574B    2694B     2780B     2832B     2956B
3712   PASSA
3744   PASSB    3742B
4396   PASSC     669C    936C      2353      2391      3040
3708   PASSD
4391   PAT       967     1019
4043   PCPV     4054B
2490   PCRC1    2446C    2460
2489   PCRC2    2448C    2469      2477      3664
2493   PCRC3    2450C
2491   PCRC4    2452C    3492
2475   PCRC5    2461B    2470B
2485   PCRC7
2488   PCRC8    2486B
2438   PCRC-X    924B
2439   PCRREP   2505B
4393   PFLAG     799C    826C      6810
6806   PLB      1118B    3694B
6813   PLB1     6811B
6818   PLB2     6815
6819   PLB3     6814C
4036   PPAR     4023B
3171   PRCV     3004C    3746C
3380   PREV      842B    3048B
3394   PREV1    3392B
3393   PREV2    3389B
3422   PREV3    3408B    3414B
3415   PREV4    3436B
3383   PREV5    3381B
3432   PREV6    3428B
1323   PRLNLP
1388   PRO-A    1325
1390   PRO-B
1392   PRO-C
1394   PRO-D
1396   PRO-E
```

```
1398  PRO-F
1400  PRO-G
1402  PRO-H
1404  PRO-I
1406  PRO-J
1408  PRO-K
1410  PRO-L
1412  PRO-M
1414  PRO-N
1416  PRO-O
1418  PRO-P
1420  PRO-Q
1422  PRO-R
1328  PRT-1
1346  PRT-2     1343B
1322  PRT-GA     885B
1325  PRT-LP    1372B
3108  PSB       3062
 935  PSPT       907      909      914      916      918      929      931B
4048  PVL       4062B
4364  PVLU      3960C    3968     3969C
3956  PVLU2     3953B
4392  QFLG       798C     822C     927
 798  QSTR       928B
4407  R10       2475
4405  R400      2362
4406  R85       2408
3192  RAC       3729C
3287  RADD1     3737C
3294  RADD2     3763C
3301  RADD3     3776C
3308  RADD4     3788C
2941  RADTBL    2896     2907
2386  RAN-RD     919B
4374  RANGE      700C     712     2648     2657     2754C    2757     2901C    2915     3139C    3321
4977  RC5       4891     4892     4896     4949     4950     4954
5365  RC88      5352     5353     5354
5396  RC89      5333     5334     5342     5377     5378     5380
5215  RC90
5295  RC91      5185     5186     5187     5212     5213     5217
5307  RC92      5166     5167     5168
5543  RC93      5535     5536     5537
5246  RCCPD1    5283     5284     5286
5253  RCCPD2    5269     5270     5271
3514  RCEND     3501B
 724  RCK
5315  RCNXT     5310     5311     5313
3289  RCON1     3290     3754C
3296  RCON2     3297     3769C
3303  RCON3     3304     3782C
3310  RCON4
4529  RCV1      4506     4507     4508
4939  RCV10     4826     4827     4828     4835     4962     4963     4965     4967     4968     4969
5146  RCV11     5111     5112     5114
5074  RCV12     5061     5062     5064
4558  RCV2
4582  RCV3
4616  RCV4      4602     4603     4605
4644  RCV4A     4630     4631     4634
4671  RCV5      4658     4659     4661
4686  RCV6      4673     4674     4675
4705  RCV7      4692     4693     4694
4735  RCV8      4719     4720     4721
4881  RCV9      4848     4849     4855     4864     4865     4867     5287     5288     5289
1789  RCV-EA     902B
1816  RCV-EB     904B
1843  RCV-EC     906B
1870  RCV-ED     908B
2241  RCV-EN     913B
2276  RCV-EO     915B
2311  RCV-EP     917B
3184  RCVRES    3708C
2406  RD3       2401B    2403B
2425  RD4       2398C    2404C
4220  RDATA     3395B
5239  RDLAY     4872     4873     4874
4230  RDTA1     4222C
4232  RDTA2     4226C
4231  RDTA3     4224C
 669  RESTRT     662      829B    6808B
2830  RET       1454B    1496B    1536B    1728B    1801B    1828B    1855B    1882B    1957B    2253B
                2288B    2323B
2834  RET1      2830     2937B
2907  RET10     2905B
2885  RET11     2872B    2877B    2880B    2882B
2884  RET13     2874B
2860  RET14     2855B    2932B    2934B
2926  RET15     2924B
2840  RET2      2836B
2873  RET3      2870B
2887  RET4
2889  RET5      2863B
2896  RET6      2891B
2935  RET7      2930B
2922  RET8      2868B    2886B    2888B    2917B
2837  RET9      2835B
2864  RETX
3398  REVLOC    3394C
 688  RFRIQ      685B
4428  RHCFW1    3411
4429  RHCFW2
4485  RPRT      3410
3939  RPVLU     2601B    2634B    2986B    2995B    3020B    3029B    3074B    3083B
5670  RREV      5623     5624     5625
4397  RRNG      2793     3910C    3913
3288  RRNG1     3744C
3295  RRNG2     3766C
3302  RRNG3     3779C
3309  RRNG4     3791C
 699  RST        679B     733B     737B
2626  RSTLST    2619B
6801  RTB        768C     770     2648     2667     2680     2771     3287     3294     3301     3308
                3396     3412     3413C    3416     3420C    3422     3432     3444     3450     3854
```

```
                 4388
3097  RTSFLG     3058C     3069C     3145      3151C
 812  RUT1        808B      819
 836  RUT4        810B      813B      817B
 802  RUTBG       807       823B      827B      832B
 824  RUTBGA      821B
 828  RUTBGB      825B
 819  RUTEND      815B
 720  SALT        774B
6830  SAV        3116C     3357      3544C     3644      3662C     3676
6824  SAV1       3807C     3820      4128C     4135C     4156      4178C     4206      4220C     4241C     4249
                 4306C     4318      4327C     4351
6825  SAV2       3466C     3474      3839C     3849      3863C     3872      3881C     3908C     3921      4257C
                 4264      4284C     4300
6826  SAV3       3978C     3984
6827  SAV4       3939C     3972
6828  SAV5       3989C     3997C     4058      4122
6829  SAVMAJ     1111C     1115      2575C     2580      2695C     2700      2781C     2787      2831C     2837
                 2957C     2963
4178  SDATA      1128B     3221B     3229B
6800  SDB         955       967       981      1007      1019      1033      1059      1085      1251      1274
                 1308      1367      1463      1467      1504      1508      1544      1548      1624      1737
                 1741      1772      1810      1837      1864      1891      1966      1970      2262      2266
                 2270      2297      2301      2305      2332      2336      2340      2362      2376      2408
                 2424      2475      2497      2665      2679      3404      3468      3485      3517C     3519C
                 3521C     3523C     3545      3673
4419  SEC         853C     3932C
4257  SETLCT     1145B     1148B     1154B     3164B     3942B     3957B     3963B     4025B
5805  SFRM       6480      6481      6482
5872  SFRM1      5856      5857      5859
5883  SFRM2      5834      5835      5843      5868      5869      5874
5841  SFRM3      5818      5819      5825
3455  SIT12      3449B
3436  SIT13
3450  SIT14      3443B
4395  SPACE      4050
2825  SPDMSG     2808
2357  SPED1      2354B
2362  SPED2      2358B     2380
2797  SPED3      2795B
2350  SPED-W      861B
4412  SPEED       865      2400      2818C     2922
4362  SPFLG       869C      872C      921
4189  SPRG1      4181C
4190  SPRG2      4184C
4197  SPRG3      4236B
4182  SPRG4
4188  SPRG5      4182C
4206  SPRG7
2780  SPTS       2367B
2783  SPTS1      2822B
2790  SPTS2      2786B
2787  SPTS3      2784B
4360  STAT       4153C
5442  STAT1      5427      5428      5430
5468  STAT2      5453      5454      5456
5494  STAT3      5479      5480      5482
5520  STAT4      5505      5506      5508
5531  STAT5      5526      5527      5528
5416  STATER     5229      5230      5231      5241
2948  STBL       2841      2860C
6766  STDLY      6770      6771      6772
3630  STDTA      3625B
 850  STLOOP      666B      843B      856
 662  STOP        674C      851C      933C      934B      937B
3459  STPT       3447
4402  STPTR
 665  STRT        667      6831
1290  SUP-G       883B
3189  TACC       3733C
1428  TCBBSZ     1332      1577      2661      3495      3629
3590  TCBP       3583B
3601  TCBP1      3596B     3598B
6198  TDAT10     5997      5998      6003
6377  TDAT11     6331      6332      6333      6346      6347      6348
6286  TDAT12     6224      6225      6226
6237  TDAT13     6590      6591      6592      6594
6001  TDAT2      5984      5985      5987
6036  TDAT3      5956      5957      5960      5969      5970      5975      6579      6580      6581
6065  TDAT4      6057      6058      6059
6024  TDAT5      6012      6013      6015
6306  TDAT6      6506      6507      6508
6425  TDAT7      6380      6381      6389
6477  TDAT8      6435      6436      6437      6461      6462      6463
6088  TDAT9      5937      5938      5947      6099      6100      6101      6105      6106      6115
5925  TDATA
4386  TEMP       2812C     2813      3402C     3440      3667C     3681      3683      4048C     4049
6387  TEND
1568  TERM-J      895B
3881  TEST       3354B     4197B
3280  TESTM      3244B     3250B     3259B     3270B
3244  TESTT
3330  TESTU
3324  TESTV      3322B
3344  TESTX      3345B
3314  TESTY      3291B     3298B     3305B
3893  TESTZ      3901B
3909  TESTZ1     3894B     3896B
3916  TESTZ2     3914B
1962  TFLG       1921C     1937
1912  TFR-M       910B
3356  TIMOUT     3324C
4306  TKSEC      2801B
4400  TLOC       4309C     4315      4317C     4333
1913  TLUP       1975B
4462  TMPSTR      705       707C      712       714       717       787      3380C     3455
2959  TON1       3095B
2976  TON10      2974B
2963  TON2       2960B
2969  TON3       2961B
3049  TON6       3039B     3041B
3064  TON8       3089B
3091  TON9       3060B     3066B
```

```
 943  TON-AA     857B
6410  TONER     6248      6249      6252      6257      6258      6261
2956  TONTST     947B
4385  TPR        763C      767       803       804      1134C     1141      1181C     1189      2805C     2807
                2811C     2813      2848C     2857C     2864C     2903      2913C     3046C     3047      3427C
                3429      3434C     3435      3446C     3448      4038C     4049      4051C     4057C
1257  TRAN1     1259B     1278
1245  TRAN-F     881B
1661  TRM-B
1667  TRM-C
1673  TRM-D
1679  TRM-E
1685  TRM-F
1691  TRM-G
1697  TRM-H
1703  TRM-J
1623  TRMAC     1599C
1621  TRMFLG    1594C
1617  TRML6     1583C     1612
1618  TRML7     1572C
1619  TRML8     1574C
1570  TRMMLP    1629B
1625  TRMRNG    1606C
1655  TRMTBL    1570
1967  TRNG1     1923C
1971  TRNG2     1925C     1939      1946C
4464  TSA2
3220  TST1A     3166B
2655  TST1C     2650B
2672  TST1D     2656B     2669B
2687  TST1E     2673B
2685  TST1F     2683B
2679  TST1G     2685B
4418  TTOT       854C     3926      3927C     3931C
5944  TXYZ      6075      6076      6078      6080      6081      6082
3190  UFMCNT    3731C
1754  UND1-K     900B
2772  UND2      2752B     2768B     2770B
2764  UND3      2759B
3926  UPTM      3919B     4141B
3933  UPTM1     3930B
3679  VERT      3675B     3687B
3239  XADD1     2610      2941      3735C
3252  XADD2     2618      2942      3256C     3748C     3761C
3264  XADD3     2943      3268C     3774C
3275  XADD4     2944      3279C     3786C
1964  XCON      1927C     1944C
3241  XCON1     3243      3248      3753C
3254  XCON2     3258      3749C     3768C
3266  XCON3     3269      3750C     3781C
6572  XDLAY     6020      6021      6022
6584  XDLAY1    6229      6230      6231      6239
4425  XHCFW1    3419
4426  XHCFW2
4398  XLOOP      800C      816C     3152
5617  XMIT21    5594      5595      5596
4484  XPRT      3415
5738  XREV      5661      5662      5663      5714      5715      5716
5700  XREVX     5729      5730      5731
3240  XRNG1     2589      2716      2745C     2764      3132      3739C
3253  XRNG2     3137      3747C     3765C
3265  XRNG3     3778C
3276  XRNG4     3790C
5764  XSTAR     5757      5758      5760
3196  XTEMP     2584C     2981C
3188  XTMSK     3129C
6488  XUR       6303      6304      6308
4801  XYZ       5407      5408      5409      5418
 644  ZERO       661      6799      6802
      ZHCOMM     650
      ZHIAFB     647
      ZHISAZ     647      4139C
      ZHNTSA     645
      ZHPFR      650      3882      4138
      ZHRTCC     648      2799      3341C     3344      3889C     3895      3918      4140C     4150
      ZHRTCI     649      3338C     3884C     4143C
      ZHRTCL     649      3340C     3886C     4145C
      ZHTH15     645       719C
      ZHWDTC     648
      ZV$BKF     652      6807
      ZV$BRK     652      6806B
      ZV$C      2648B     2757B     2915B
      ZV$ER     4013B
      ZV$F       705B      967B     1019B     2362B     2841B     3854B
      ZV$FI      646      2406B
      ZV$FR      646      2408B     2475B
      ZV$HR      645
      ZV$IA      803B
      ZV$ID      687B
      ZV$IH      712B
      ZV$IZ      786B
      ZV$PCH     651       732B
      ZV$QC      653       686B      711B      802B
      ZV$RD      646       672B
      ZV$T       647       771B     3401B     3410B     3415B     3430B     4016B
      ZV$TC      653       736B      769B      831B      838B      935B     2806B     2808B     3043B     3447B
                6815B
      ZV$TD      653       770B     2807B     2809B     2813B     3047B     3429B     3435B     4017B
      ZV$TH      772B
      ZV$THZ     653      3448B     4049B
      ZV$TTY     646       930
 646 LABELS
3318 REFERENCES
6831 RECORDS
   0 U FLAGS
   0 M FLAGS
  67 N FLAGS
6 CROSS REF VERSION L - 24 SEPT, 1976
RS LINKER VERSION 5.05 07/21/78  1809.4 EDT FRI
LINK MAP FOR DCMS3
 START        0100
 LOW          0000
 HIGH         29A6
```

```
  CURRENT        29A7
*LOC DEFS
   ZHCOMM        0000
  *DCMS3         0000   REV A
   ZHPFR         0000
   ZHTSA         0002
   ZHNTSA        0010
   ZHRTCI        0014
   ZHRTCC        0015
   ZHRTCL        0016
   ZHWDTC        0017
   ZHMERC        001F
   ZHIAFB        0020
   ZHTH29        0063
   ZHTH28        0064
   ZHTH27        0065
   ZHTH26        0066
   ZHTH25        0067
   ZHTH24        0068
   ZHTH23        0069
   ZHTH22        006A
   ZHTH21        006B
   ZHTH20        006C
   ZHTH19        006D
   ZHTH18        006E
   ZHTH17        006F
   ZHMEMP        006F
   ZHTH16        0070
   ZHLERR        0070
   ZHTH15        0071
   ZHNRES        0071
   ZHTH14        0072
   ZHPMEM        0072
   ZHTH13        0073
   ZHP-OP        0073
   ZHTH12        0074
   ZHTH11        0075
   ZHTH10        0076
   ZHTH9         0077
   ZHTH8         0078
   ZHTH7         0079
   ZHTH6         007A
   ZHOVFL        007A
   ZHTH5         007B
   ZHOP-N        007B
   ZHTH4         007C
   ZHTH3         007D
   ZHSC-N        007D
   ZHTH2         007E
   ZHTRC         007E
   ZHTH1         007F
   ZHMCL         007F
   ZHISAZ        0080
   ZHIVBS        0080
   ZHTVBS        0080
  *ZV$TH         2083
   ZV$TD         20B8
   ZV$TH         2083
   ZV$THZ        20AB
  *ZV$F          20D3
   ZV$F          20D3
  *ZV$IH         20E1
   ZV$ID         20E6
   ZV$IH         20E1
   ZV$IAD        20EB
   ZV$--2        2103
   ZV$--3        2115
  *ZV$PCH        217A
   ZV$PCH        217A
  *ZV$T          227C   REV. 5.0
   ZV$QC         2299
   ZV$TC         2285
   ZV$T          227C
   ZV$Q          228E
  *ZV$IA         22AD   REV. 7
   ZV$IA         22B0
   ZV$ABF        2361
   ZV$ARG        235F
   ZV$--1        231C
   ZV$IAV        22AE
  *ZV$FR         236C
   ZV$FI         238E
   ZV$FR         236C
   ZV$FS         23B1
   ZV$FRA        23BE
   ZV$FRX        23BF
   ZV$FRR        2383
   ZV$FRB        23C0
   ZV$FRM        23BD
  *ZV$C          23C3   REV. 5
   ZV$C          23C3
   ZV$CO         23E6
  *ZV$ER         23F7   REV. 5.0
   ZV$ER         23F7
   ZV$TA         2423
   ZV$--0        240A
  *ZV$BRK        2467
   ZV$BRK        2467
  *ZV$GP         2481
   ZV$GP         2481
   ZV$--4        24A1
  *ZV$HA         24AD
   ZV$HA         24AD
   ZV$HZ         24B7
   ZV$HS         24B2
  *ZV$HD         24E6
   ZV$HD         24E6
  *ZV$RD         2518   REV. 7
   ZV$RD         2518
   ZV$IZ         2552
   ZV$TTY        252B
   ZV$BKF        2540
   ZV$SV1        26ED
   ZV$SV3        270D
   ZV$AF         2529
   ZV$SV2        26FD
   ZV$OTP        25BF
   ZV$TID        252A
   ZV$CF2        2534
   ZV$TK         2530
   ZV$RAR        2531
   ZV$STI        2535
   ZV$RCC        2536
   ZV$BUD        252C
   ZV$OLB        2538
   ZV$RCB        2539
   ZV$NSR        253D
   ZV$STR        253B
   ZV$BKS        253F
   ZV$HR         2547
   ZV$LR         2544
   ZV$DAT        2527
   ZV$HM         258E
   ZV$HRU        2541
   ZV$HRL        2542
   ZV$LRU        2543
   ZV$LRL        2544
   ZV$HBD        2545
   ZV$CFI        2533
   ZV$--5        254A
   ZV$RMD        2528
   ZV$MCP        2546
   HIBAUD        2545
   ZV$RAW        2532
   ZV$RDT        2749
   ZV$CTL        252F
   ZV$BI         266A
   ZV$TST        279F
   ZV$MDC        2773
   ZV$R99        2971
   ZV$ISA        254D
   ZV$UIH        2548
   ZV$ZRO        25CC
   ZV$BSH        25CE
   ZV$CPU        252E
   ZV$R50        25AC
   ZV$R60        25B7
   ZV$RT         28AE
   ZV$ALL        252D
  *MLCHPG        2976   T+V
   MLCHPG        2976
   ENDCHP        29A7
 *UNLINK MODULE(S)
  ZV$QC
  ZV$ID
  ZV$TC
  ZV$TD
  ZV$IZ
  ZV$FI
  ZV$THZ
```